TDK Dolgozat

Bizonytalanság kvantifikálása bayesi neurális háló alapú módszerekkel

---

Vetró Mihály
NVI265

Konzulens:
Dr. Hullám Gábor

2020

Tartalomjegyzék

1. Bevezetés

Az elmúlt évtizedek technológiai fejlődésének köszönhetően az adatok alapján történő tanulás és következtetés szinte alapértelmezett módszerré vált a becslési problémák megoldásához, tekintve, hogy a legtöbb alkalmazásban így jelentősen jobb prediktív teljesítményt lehet elérni, mint pusztán előre meghatározott szabályok alkalmazásával. A mára nagyrészt megoldottnak nyilvánított erőforrásigényből és adathiányból következő problémák mellett azonban van egy harmadik, jóval kevesebbet emlegetett probléma is együtt jár ezen módszerekkel, mégpedig az így készült pediktív modellek megbízhatósága. Ez főként abból adódik, hogy a legtöbb paraméter alapú tanuló modellt komplexitása miatt "fekete doboznak" (*black box*) tekintjük, tehát pusztán szerkezetük és paramétereik ismeretében nem tudunk semmilyen garanciát adni arra, hogy a modell konzisztens módon megfelelő predikciót fog adni a bemeneti tér teljes egészében. Ez legfőképp abból adódik, hogy a legtöbb problémánál a bemeneti tér puszta méretéből fakadóan praktikus módon nem tudjuk a modell viselkedését a bemeneti tér teljes egészében megfigyelni. Ezen megbízhatósági probléma megoldása végett született meg a bayesi neurális hálózatok (BNN), vagy általánosabban bayesi tanuló modellek koncepciója, amely modellek képesek kezelni a kimenet bizonytalanságát, beleértve az olyan bemenetek is, amelyekhez hasonlót még nem látott a modell a tanítási folyamat során. Az ilyen modellek bayesi megközelítése alapvetően két különböző irányból történhet: (1) megpróbálhatjuk direkt módon, előre meghatározott alakban és a priori eloszlás mellett megkeresni a modell változói felett értelmezett a posteriori eloszlást, vagy (2) valamilyen kontrollált véletlen folyamat felhasználásával próbálunk mintákat venni az amúgy ismeretlen alakú a posteriori eloszlásból. Ezen két megközelítés közül az első (1) kategóriába eső módszereket variációs (*variational*), a második (2) kategóriába esőeket pedig Monte Carlo módszereknek nevezzük. A dolgozatomban ismertetni fogom mindkét megközelítés előnyeit és hátrányait, a variációs módszerek közül külön figyelmet fordítva a Bayes-by-Backprop (BBB) [Blu+15] nevű módszerre, a Monte Carlo módszerek közül a Stochastic Gradient Langevin Dynamics (SGLD) [Kor+15] nevű mintavételezési módszerre. Ezek mellett bemutatok egy új, távolságregularizációs modellegyüttesen alapuló közelítő módszert (DREAM: Distance Regularized Ensemble Approximation Method), amely az ismertetett módszerekhez képest jobb teljesítményt nyújt a bizonytalanság becslése terén a vizsgált szintetikus és valós adatokon egyaránt, legalább ugyanolyan jó prediktív teljesítmény és erőforrásigény megtartása mellett.

2. A bizonytalanság típusai

Egy adott gépi tanulási probléma mellett fellépő bizonytalanság kategorizálására több nézőpont is létezik, amelyek közül itt a bizonytalanságot annak forrása szerinti kategóriákba sorolását fogom bemutatni, azon belül is az aleatorikus és episztemikus bizonytalansági típusokat.

2.1. Aleatorikus bizonytalanság

Az aleatorikus bizonytalanság közvetlenül a mérés bizonytalansága, tehát egy ugyanazon körülmények között többször elvégzett kísérlet eredményében tapasztalható variancia, az adatot, és így közvetve a modellt is jellemzi. Tekintve, hogy ezen típusú bizonytalanság forrása közvetlenül az az amúgy ismeretlen "modell", amely a valóságban a tanításhoz használt adatainkat előállította, és amelyet a tanuló modellünkkel szeretnénk közelíteni egy gépi tanulási probléma megoldása során, így ezen bizonytalanság csökkentésére valós körülmények között nem sok lehetőségünk van. Ettől függetlenül természetesen megfigyelhetjük, hogy a bemeneti térben egymáshoz közel eső bemenetekhez tartozó elvárt kimenetek mennyire térnek el egymástól, amely (a bemenetek közelségével normalizálva) önmagában is jó metrikaként szolgálhat az aleatorikus bizonytalanság mérésére.

2.2. Episztemikus bizonytalanság

Az episztemikus bizonytalanság ezzel szemben a megfigyelések, és így a begyűjtött adatok hiányosságából adódó bizonytalanság, amely az episztemikus bizonytalanságnál sokkal közvetlenebb módon jellemzi a modellt. Ahogyan [HW20] 2.2. fejezetében is olvasható, ebbe a kategóriába tartozik továbbá a hipotézisterünk (vagy esetünkben paraméterterünk) limitáltságából, illetve az abban megvalósított kereső algoritmus tökéletlenségéből adódó bizonytalanság is, amely a legtöbb esetben szinte elkerülhetetlenül egy szuboptimális hipotézishez (vagyis paraméterezéshez) vezet. Ez lényegében azt jelenti, hogy a vizsgált problémára nyújtható tökéletes megoldást valós alkalmazás esetén a hipotézisterünk (paraméterterünk) szinte biztosan nem tartalmazza, illetve a tanuló, hipotézisünket (paramétereinket) optimalizáló algoritmus sem fogja megtalálni a legjobb, még a hipotézisterünkön belülre eső megoldást. Ezekből adódóan viszont fontos észrevennünk, hogy mivel ezen bizonytalanság nem a megfigyelt világ entrópiájából, hanem közvetlenül a megfigyelési és modellalkotási folyamatunk hiányosságaiból adódik, így ezen folyamatok (vagy módszerek) optimalizálása által ezen típusú bizonytalanság értelemszerűen csökkenthető.

2.3. A bizonytalanság típusainak megkülönböztetése paraméter alapú tanuló modellekben

Ahogyan az előző bekezdésben beláttuk, a bizonytalanság két típusa közül (amelyek valós esetben egyaránt jelen vannak) csupán az episztemikus bizonytalanság az, amelyet a modell módosítása által tudunk csökkenteni. Ebből adódóan érdemes lehet vizsgálni, hogy lehetséges-e az összesített bizonytalanság ezen két típusra történő közvetlen numerikus felbontása. Erre [HW20] 4.3. fejezetében említ egy módszert, amely szerint hogyha meghatározzuk a modellünk a posteriori prediktív eloszlásának entrópiáját ($H[p(y|\boldsymbol{x})]$), amely a modell teljes bizonytalanságának felel meg, illetve emellett a fix hipotézissel (vagyis paraméterezéssel) rendelkező modellek ugyanezen eloszlása fölött vett entrópiájának a paraméterek *a posteriori* eloszlása szerint súlyozott várható értékét ($\boldsymbol{E}_{p(\theta|X_{train},Y_{train})}H[p(y|\theta,\boldsymbol{x})]$), amely lényegében az aleatorikus bizonytalansággal egyenértékű, akkor a kettő különbsége az episztemikus bizonytalanságot, illetve egyúttal a kimenet és a hipotézis (modellparaméterek) közötti kölcsönös információ értékét adja:

$$H[p(y|\boldsymbol{x})] - \boldsymbol{E}_{p(\theta|X_{train},Y_{train})}H[p(y|\theta,\boldsymbol{x})] = I(y,\theta) \tag{1}$$

3. A bizonytalanság mérése

Alapvetően bizonytalansági metrikának nevezhetünk egy modell esetén egy olyan értéket, amely közvetlen (pozitív vagy negatív irányú) korrelációban van a modell várható tévedésével[1]. Habár ezen metrikát végső soron mindig a végleges hipotézisünk mibenléte határozza meg, ezen pontos hipotézis kialakulására, és ezáltal közvetetten a bizonytalanság mértékére több tényező is befolyást gyakorol.

[1]Itt a tévedést az abszolút valósághoz képest értjük, tehát a tanító adat valóságtól való eltérése is tévedésnek számít.

3.1. A metrikák pontossága

Alapvetően fontos észrevennünk, hogy bármilyen, bizonytalanságot leíró értéket is veszünk fel, az továbbra is egy, az eddig megfigyelt adatokból és egyéb hipotézisből, következtetésből adódó érték lesz, amelyek tökéletlenségéből adódóan a bizonytalanság mérésére bevezetett értéknek is lesz bizonytalansága. Így tehát erősen indokolt ellenőrízni, hogy az általunk meghatározott bizonytalansági metrika mennyire jól írja le a valóságot. Erre talán a leginkább kézenfekvő módszer, hogyha megvizsgáljuk, hogy a rendelkezésünkre álló tesztadatokon mennyire erős a korreláció a modell tévedése és az általa adott bizonytalanság-érték között. Habár ezen validációs módszer is erősen függ a rendelkezésünkre álló adatoktól, alkalmazásával legtöbb esetben jó képet kaphatunk arról, hogy a modellünk képes-e a bizonytalanság megfelelő becslésére valós esetben.

4. A paraméter alapú tanuló modellek bayesi megközelítése

Ha a modell kimenetének bizonytalanságát szeretnénk meghatározni, akkor ezen kimenetre már nem egy skalár értékként, hanem egy adott eloszlással rendelkező valószínűségi változóként kell gondolnunk minden lehetséges bemenetre. Ezen kimenet eloszlása közvetlenül a bemenettől, illetve a hipotézistől, utóbbi által pedig közvetetten a tanító adatoktól függ, tehát legegyszerűbb formában az alábbi módon írható fel:

$$p(y|x, X_{train}, Y_{train}) \tag{2}$$

Ahol x és y a pillanatnyilag vizsgált bemenet-kimenet pár, X_{train} és Y_{train} pedig a modell tanításánál felhasznált bemenetek és a hozzá tartozó elvárt kimenetek halmaza. A tanító adatok alapján felállított hipotézist a modell paraméterei (θ) reprezentálják, amelyet bevezetve a kimenet eloszlását az alábbi módon bonthatjuk fel:

$$p(y|x, X_{train}, Y_{train}) = \int p(y|x, \theta)p(\theta|X_{train}, Y_{train})d\theta \tag{3}$$

Az integrálon belüli szorzatban a $p(y|x, \theta)$ tag kiszámítása x és θ ismeretében triviális, ugyanis a kimenetet ezen két változó egyértelműen meghatározza. Ebből adódóan a kimeneti változó eloszlásához a paraméterek a posteriori eloszlását kell meghatázornunk, a tanító adatok függvényében. Sajnos ezen eloszlás meghatározása egzakt módon gyakorlati problémák esetén nem lehetséges, viszont ezen esetekre számos közelítő módszer létezik, ahogyan a későbbiekben látni fogjuk.

5. Ismert módszerek a paraméterek eloszlásának közelítésére

Ahogyan a bezetőben is láthattuk, a modell paraméterei fölött értelmezett eloszlás közelítését szolgáló módszerek alapvetően két csoportba sorolhatók: (1) vannak a variációs módszerek, amelyek egzakt módon, egy adott formában keresik a paraméterek a posteriori eloszlását, illetve (2) a Monte Carlo módszerek, amelyek egy

ismeretlen alakú a posteriori eloszlásból végeznek mintavételezést. Ebben a fejezetben ismertetni fogok néhány, már létező módszert mindkét említett kategóriából, kiemelve azok előnyeit és hátrányait elméleti és gyakorlati szempontból egyaránt.

5.1. Variációs módszerek

Általánosan a variációs módszerekről elmondható, hogy a paraméterek fölötti a posteriori eloszlást egy meghatározott alakban keresik, egy szintén adott - és rendszerint az a posteriorival azonos - alakú a priori eloszlásból kiindulva. Ezen módszerek többségéről elmondható, hogy alkalmazásukhoz (főként az a priori és a posteriori eloszlások meghatározása miatt) a megszokottnál valamivel több tervezőmunka szükséges, és a tanítás is egy jóval összetettebb folyamattá válik, tekintve, hogy a modell paraméterei helyett az ezen paraméterek fölött értelmezett eloszlás változóinak (vagyis paramétereinek) keressük az optimális értékét. Mindazonáltal ezen módszerek előnyei közé sorolható, hogy a normál (nem variációs) modellhez képest a paraméterek számának növekedése, és így a tanítás futásideje is csupán az eredeti skalárszorosára növekszik. Így például hogyha normál eloszlást illesztünk a paraméterekre, akkor minden paraméternek lesz egy várható értéke és szórása, amely változónként összesen két értéket jelent. Ezáltal a modell variációs változatának a paraméterszáma és (várhatóan) a tanítási ideje is az eredeti duplájára nő.

5.1.1. Bayes-by-Backprop

Az egyik talán leginkább kézenfekvő variációs módszer a Google mérnökeinek [Blu+15] cikkében bemutatott Bayes-by-Backprop nevű eljárás. Ennek lényege, hogy minden

1. Sample $\epsilon \sim \mathcal{N}(0, I)$.
2. Let $\mathbf{w} = \mu + \log(1 + \exp(\rho)) \circ \epsilon$.
3. Let $\theta = (\mu, \rho)$.
4. Let $f(\mathbf{w}, \theta) = \log q(\mathbf{w}|\theta) - \log P(\mathbf{w})P(\mathcal{D}|\mathbf{w})$.
5. Calculate the gradient with respect to the mean

$$\Delta_\mu = \frac{\partial f(\mathbf{w}, \theta)}{\partial \mathbf{w}} + \frac{\partial f(\mathbf{w}, \theta)}{\partial \mu}. \tag{3}$$

6. Calculate the gradient with respect to the standard deviation parameter ρ

$$\Delta_\rho = \frac{\partial f(\mathbf{w}, \theta)}{\partial \mathbf{w}} \frac{\epsilon}{1 + \exp(-\rho)} + \frac{\partial f(\mathbf{w}, \theta)}{\partial \rho}. \tag{4}$$

7. Update the variational parameters:

$$\mu \leftarrow \mu - \alpha\Delta_\mu \tag{5}$$
$$\rho \leftarrow \rho - \alpha\Delta_\rho. \tag{6}$$

1. ábra. A Bayes-by-Backprop algoritmus leírása.

paraméter fölött, egymástól függetlenül értelmezünk egy-egy, várható értékével és szórásával egyértelműen meghatározható normál a priori eloszlást, majd ezen eloszlások paramétereire a szerzők által kidolgozott módszer segítségével minden iterációban kiszámoljuk a gradienst, majd abból a paramétermódosítás értékét is. A tanítás folyamatát az 1. ábrán látható algoritmus írja le. Ezen módszer talán legnagyobb előnye, hogy a paraméterek a posteriori eloszlása az a priori eloszlástól csupán a pontos paraméterezésében különbözik, így tehát normál a priori eloszlás esetén az a posteriori eloszlás is normál eloszlás lesz. Ebből adódóan azonban a módszer teljesítménye erősen függ attól, hogy az általunk feltételezett eloszlás-típus mennyire közelíti jól a paraméterek valós (amúgy ismeretlen) eloszlását. Emellett ezen módszer talán legnagyobb hibája, hogy az egyes változók egymástól teljesen függetlenek, amely függetlenség a változók valós eloszlására szinte biztosan nem teljesül.

5.2. Monte Carlo módszerek

A variációs módszerekkel ellentétben az ezen csoportba tartozók merőben eltérő megközelítést képviselnek: egy meghatározott formájú eloszlás helyett ismeretlen formában próbálják közelíteni a paraméterek a posteriori eloszlását valamilyen véletlen folyamat segítségével, amelyből aztán ugyanezen folyamat segítségével mintákat vesz.

5.2.1. SGLD

A Monte Carlo módszerek közül talán a legegyszerűbb, és egyben legkézenfekvőbb, a [Kor+15] által bemutatott Stochastic Gradient Langevin Dynamics (SGLD) módszer. Ennek lényege, hogy a paraméterek a posteriori eloszlásából egy korlátozott módon konvergens "véletlen sétával" veszünk mintákat. A megvalósításhoz így lényegében mindössze annyit kell tennünk, hogy a normál tanítás során használt, MAP (maximum a posteriori) becslést adó tanító algoritmus paraméterfrissítése során számolt gradienséhez 0 várható értékű, és rögzített szórással rendelkező Gauss-zajt adunk az alábbi módon:

$$\theta^{k+1} = \theta^k + \lambda \left(\nabla_\theta logp(\theta^k) + \nabla_\theta \sum_{i=1}^{N} logp(y_i|x_i, \theta^k) \right) + \eta^k \tag{4}$$

ahol a θ^k a k-adik iteráció után előálló paramétervektor, $logp(\theta^k)$ lényegében egy regularizációs tényező, $\sum_{i=1}^{N} logp(y_i|x_i, \theta^k)$ a tanító adatokon számolt hiba, λ a tanulási tényező, η^k pedig a k-adik iterációhoz, fix Gauss-eloszlásból mintavételezett additív zaj. A módszer futása során néhány kezdeti "*burn-in*" iterációt követően a θ^k paramétervektor által felvett érték korlátozott mértékben fog a pillanatnyilag legközelebbi lokális optimumhoz tartani, így tehát értékei egy, a paramétertér lokális optimumai körül tett "véletlen sétát" reprezentálnak majd, amennyiben megfelelően választottuk meg az η^k normál eloszlású zaj szórását. Így tehát a θ^k által felvett értékeket tekinthetjük a $\boldsymbol{\theta}$ változó a posteriori eloszlásából vett mintáknak.

5.2.2. Anchored ensembling

A [Pea+20] által bemutatott Monte Carlo módszer alapja, hogy egy adott problémára egymástól függetlenül inicializált modelleket tanítunk, amely tanítás során alkalmazott veszteségfüggvényben a veszteség értékéhez hozzáadjuk a modellparaméterek kezdeti értéktől való pillanatnyi távolságát a paramétertérben. Így a modellek veszteségfüggvénye a következő:

$$Loss_j = \frac{1}{N}||y - \hat{y}||_2^2 + \frac{1}{N}||\Gamma^{1/2} \cdot (\theta_j - \theta_{anc,j})||_2^2 \tag{5}$$

ahol $\frac{1}{N}||y - \hat{y}||_2^2$ egy egyszerű átlagos négyzetes hiba, $\frac{1}{N}||\Gamma^{1/2} \cdot (\theta_j - \theta_{anc,j})||_2^2$ pedig az úgynevezett "anchor" (azaz horgony) regularizációs tényező, amely ahogyan a neve is sejteti, annál nagyobb, minél nagyobb távolságba kerül a tanító algoritmus a kiindulási ponttól a paramétertérben, ezáltal növelve a loss értékét is a kiindulási ponttól távol eső helyeken. Utóbbin belül a Γ egy regularizációs ténylezőket tartalmazó diagonális mátrix, θ_j a modell aktuális paramétervektora, $\theta_{anc,j}$ pedig ugyanezen modell kiinduló paramétervektora. Ezen módszer jelentős hasonlóságot mutat az ú.n. "swarm" optimalizációs módszerekkel, ugyanis azokhoz hasonlóan itt is véletlenszerűen mintavételezzük a paraméterteret, majd az egyes mintákat elkezdjük a gradienseik szerint korrigálni, amíg azok el nem érnek egy lokális optimumpontba, esetünkben odafigyelve arra, hogy lehetőleg az adott példány kiindulási pontjához legközelebb eső lokális optimumot találjuk meg. Ez alapján sejthető, hogy ezen módszer jelentősen nagyobb diverzitású mintahalmazt fog eredményezni az SGLD módszerből generált mintákhoz képest, feltéve, hogy egymástól megfelelően távol eső kezdőpontokat választunk a modelleknek a paramétertérben. Emellett viszont ezen módszer erőforrásigénye is várhatóan jelentősen nagyobb, tekintve, hogy minden mintavételhez be kell tanítanunk egy egyedi modellt, míg a 5.2.1 részben bemutatott SGLD módszernél egy adott "warm-up" periódust követően minden lépés egy újabb mintát eredményez. Végül pedig fontos kiemelni, hogy ezen módszernél a kezdőponttól való távolság korlátozása jelentősen ronthat a minták prediktív teljesítményén, amennyiben a kiválasztott kezdőpont közelében nincs megfelelően jó lokális optimuma a problémához tartozó veszteségfüggvénynek.

6. A mintavételezésből adódó gyakorlati problémák, és azok minimalizálása

Az 5. fejezetben ismertetett módszerek formájában láthattuk, hogy már több különböző megközelítés is létezik a modell paraméterei fölött értelmezett a posteriori eloszlás közelítésre a tanító adatok ismeretében. Fontos azonban, hogy a paraméterek eloszlásának ismerete önmagában még nem elég ahhoz, hogy megfigyeljük a kimenet eloszlását egy adott bemenet függvényében. A paraméterek eloszlásához hasonlóan a kimenet eloszlásának megfigyelése is első sorban mintavételezés útján történik. Így tehát a paraméterek a posteriori eloszlásából vett mintákon kiértékeljük az adott bemenetet, az így kapott, a paraméter-minták halmazával megegyező nagyságú kimeneti mintahalmazból pedig következtethetünk a kimenet eloszlására. Ez utóbbi következtetéskor leggyakrabban normál eloszlással közelítjük a kimenet

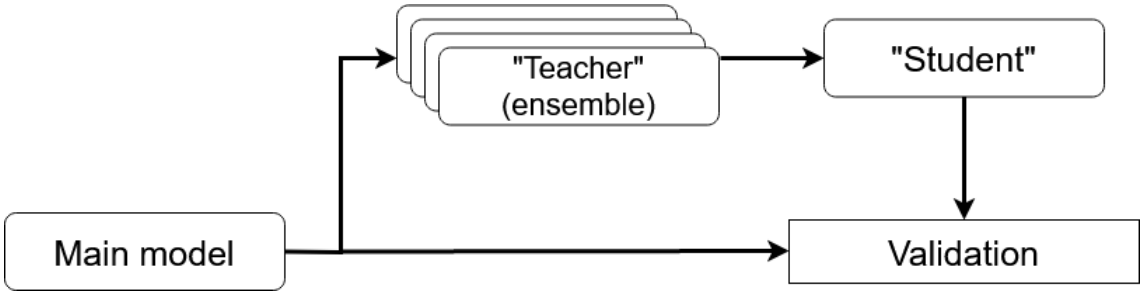
eloszlását, így tehát az eloszlás várható értéke a kimenet-minták átlaga, a szórása pedig a kimenet-minták szórása lesz. Így normál eloszlás használatakor annak várható értékét tekinthetjük a modell által adott predikciónak, szórását pedig a modell adott predikcióhoz tartozó bizonytalanságának.

6.1. Teljesítmény és a "Distilled" SGLD modell

Általánosan elmondhatjuk, hogy minél több mintát veszünk a paraméterek a posteriori eloszlásából, annál jobb közelítést kapunk a kimenet eloszlását illetően. Előfordulhat azonban, hogy főként korlátozott rendelkezésre álló erőforrás, vagy komplex, nagy paraméterszámmal rendelkező modellstruktúra esetén nincs lehetőségünk akkora méretű mintahalmazt (tehát modellegyüttest) használni a következtetés során, amely elegendő lenne a kimenet eloszlásának megfelelő közelítéséhez. Ennek megoldása végett [Kor+15] az ú.n. Distilled SGLD, modellpároson alapuló módszert javasol. Ezen módszernek a lényege, hogy az eredeti, a posteriori eloszlásból mintavételezett modellhalmaz kimenetét véletlenszerűen, a bemeneti teret egyenletesen lefedve mintavételezzük, majd az erre adott válaszok eloszlására vonatkozó értékeket (pl. normál eloszlás esetén várható értéket és szórást) rögzítjük. Ezt követően az így kialakult bemenet-kimenet párok segítségével betanítunk egy egyszerű modellt, hogy az a látott bemenetek környezetében jól közelítse az eredeti modell kimeneti eloszlásának paramétereit. Ezen tanítást ([Kor+15] javaslata szerint) az eredeti, "tanító" modell tanításával egyidőben is megtehetjük úgy, hogy minden tanítási lépést követően mintavételezzük a legfrissebb paramétervektor szerint előálló modell kimenetét, majd az így előállt bemenet-kimenet párosok segítségével végrehajtunk egy tanítási iterációt a "tanuló" modellen is. Ezen módszer segítségével a tanítás habár továbbra is számításigényesebb, és valamennyivel memóriaigényesebb egy egyszerű modell tanításnál (tekintve hogy effektive két modellt tanítunk és tárolunk el egyszerre), ám ez a megnövekedett erőforrásigény a normál mintavételezéses következtetéshez képest már nem számottevő.

6.2. Utólagos validáció konfidencia-intervallum alapján

Habár az előző bekezdésben említett Distilled SGLD módszer jó megoldást adhat a hatékonysággal kapcsolatos problémákra, semmi garanciát nem biztosít arra, hogy a bemeneti tér a tanítás során végrehajtott mintavételezés által nem lefedett részein is jó becslést ad majd a bizonytalanságra a "tanuló" modell. Ebből adódóan plusz validációs lépésként érdemes megfigyelni egy normál, MAP becsléssel előállított modell kimenetét is, majd összevetni azt a tanuló modellünk kimenetével. Amennyiben a tanuló modell kimenete egy normál eloszlás, akkor ahhoz egy elvárt valószínűséget meghatározva a várható értéke és a szórás alapján megállapíthatjuk a megfelelő konfidencia intervallumot, amelybe a kimenet a modell becslése alapján adott valószínűséggel bele fog esni. Ezt követően egy plusz validációs lépésként megvizsgálhatjuk, hogy ezen konfidenciainervallumba beleesik-e az eredeti, MAP becsléssel előállított modell kimenete. Ezen validációs módszer működését a 2. ábra foglalja össze.



2. ábra. A Distilled SGLD modell validációja.

6.3. Szintetikus tesztadatok

Az itt bemutatott módszerek ellenőrizhető teszteléséhez generáltam ugyanazon eloszlásból egy tanító és egy validációs adathalmazt, amelyek egy viszonylag egyszerűbb, 20 bemeneti változóval rendelkező regressziós problémát reprezentálnak. Emellett létrehoztam további két adathalmazt is, amelyek mindegyikét egyenletesen, véletlenszerűen generált mintákkal töltöttem fel. Ezen két adathalmaz közül az elsőt az eredeti adatokat befoglaló legkisebb hiperkockán belülről mintavételeztem (így tehát azok értékei az eredeti adatok maximum és minimum értékei közé esnek), a másodikat pedig kizárólag ezen hiperkockán kívülről, tehát abból a tartományból, amelyet az eredeti regressziós probléma bemenetei egyáltalán nem fednek le.

6.4. Eredmények a teszt adathalmazon

Az előző bekezdésben említett tesztadatokra számolt eredmények a 3. ábrán láthatók. Jól látszik, hogy összességében a tanító és a tanuló modell prediktív teljesítménye nem kifejezetten jó, illetve valamilyen anomália folytán a tanuló modell jelentősen jobb prediktív teljesítményt nyújt a tanító modellnél. Látható emellett,

		Training	Validation	Random in range	Random out of range
Teacher (ensemble)	Mean average error	33.9044	36.6405	-	-
	Mean interval size	194.5239	197.1306	245.2939	278.5275
Student	Mean average error	25.1261	26.2158	-	-
	Mean standard deviation	36.9743	38.9294	49.4761	56.3464
	Acceptance at 95% CI	0.9668	0.9666	0.2784	0.0746
	Acceptance at 99% CI	0.9826	0.9816	0.3548	0.0966

3. ábra. Az SGLD módszer teszteredményei.

hogy a tanuló modell alacsonyabb átlagos szórást produkál a valós eloszlásból vett adatokon, mint a véletlenszerűen generáltakon, viszont az itt látható különbség sem elég nagy ahhoz, hogy nagy bizonyossággal meg tudjunk húzni valamilyen határt, amely fölött már nem bízunk meg a modell jóslatában. Végül azonban fontos ész-

revenni, hogy a validációnál mért "acceptance rate"[2] a valós eloszlásból generált adatokon jóval magasabb, mint a véletlenszerű adathalmazokon, amely empirikus bizonyítékként szolgál arra, hogy a bevezetett validációs módszer az elvártaknak megfelelően működik.

7. A minták diverzitásának biztosítása

Alapvetésként el kell fogadnunk, hogy egy véletlen séta, vagy egyéb véletlen folyamat a paraméterek a posteriori eloszlásának sűrűségfüggvényét várhatóan csak a paramétertér bejárt részein fogja jól közelíteni. Ebből adódóan tehát valós esetben mondhatjuk, hogy egy véletlen folyamat minél nagyobb részét "járta be" a paramétertérnek, annál jobban fogja a belőle vett minták által közelíteni az eredeti, paraméterek fölött értelmezett a posteriori eloszlást. A paramétertér minél jobb bejárását az SGLD módszernél az additív gradiens-zaj növelésével érhetjük el, amely várhatóan jelentősen rontani fog a végeredményként kapott modell prediktív teljesítményén tekintve, hogy jóval kisebb valószínűséggel fog valamelyik lokális optimum irányába konvergálni a tanítási folyamat. Az Anchored Ensembling esetén ugyanezt a feltételt megpróbálhatjuk egymástól távol eső kiindulási paraméterek (tehát "Anchor"-ok) választásával biztosítani, azonban amennyiben az éppen választott kezdőpont közelében nincs kellően jó lokális optimum, úgy szintén a modell prediktív teljesítményének jelentős romlását okozhatjuk ezzel. Ezen probléma megoldása végett bevezettem egy új módszert, amely közvetlenül az éppen tanítás alatt álló modell veszteségfüggvényébe építi be az eddig betanított modellektől (tehát a paraméterhalmazból vett mintáktól) való távolságot additív regularizáció formájában. Ezen módszer, és altípusai a *Distance Regularized Ensemble Approximation Method* (vagy rövidítve: *DREAM*) nevet viselik. Fontos kiemelni, hogy míg az Anchored Ensembling módszer a paramétertér egy viszonylag kicsi részében, azaz a kezdőpont, vagyis a kiindulási paraméterek közvetlen közelében keres minél optimálisabb megoldást, a DREAM ezzel szemben csupán annyi megkötéssel él, hogy az újonnan vett minta minél távolabb essen a korábbi mintáktól a paramétertérben. Ebből következik, hogy mivel a paramétertér jelentősen nagyobb részében van lehetősége keresni a tanulási folyamat során, így várhatóan végeredményként optimálisabb megoldást fog találni, mint az Anchored Ensemble.

7.1. Az átlagos távolság minimalizálása

A DREAM módszer talán leginkább kézenfekvő megvalósítása, hogyha a modell veszteségfüggvényéhez hozzáadjuk az eddigi mintáktól vett átlagos távolságot egy regularizációs tényező és egy exponens kíséretében. Így tehát az $(m+1)$-dik modell veszteségfüggvénye (DREAM-AVG) az alábbi lesz:

$$Loss_{m+1} = \rho \left(\frac{1}{m} \sum_{i=1}^{m} ||\theta - \theta_i||_2 \right)^{\tau} + \lambda \sum_{j=1}^{N} logp(y_j|x_j, \theta) \tag{6}$$

[2]Tehát hogy a bemenetek mekkora hányadát fogadta el a validációs módszer az egyes adathalmazokon.

ahol a $\frac{1}{m}\sum_{i=1}^{m}||\theta-\theta_i||_2$ az összes, eddig létrehozott modelltől mért átlagos távolság a paramétertérben, ρ a regularizációs tényező, τ pedig a kötelezően negatív értékű[3] exponens, amely szintén egy hiperparaméter. Ahogyan már a formulából is sejthető, a regularizációs tényező a távolság-regularizáció erősségét adja meg a modell hibájából adódó veszteséghez képest, az exponens pedig hogy a távolság alapú regularizáció milyen ütemben csökkenjen a távolság növekedésével. Így tehát ugyanazon ρ érték mellett $\tau \in]-\infty;-1]$ esetén a regularizáció mértéke erősen lecsengő lesz a távolság növekedésével, míg $\tau \in]-1;0[$ sokkal egyenletesebben csökkenő értéket eredményez. Fontos kiemelni továbbá, hogy az itt ismertetett DREAM módszer három altípusa (ezt is beleértve) azon alapszik, hogy a modelleket egymás után, és nem párhuzamosan tanítjuk, így tehát az összes eddig betanított modell paramétervektora ismert. Habár elméletben a módszer támogatná a modellek párhuzamos tanítását is, az feltehetően egy numerikusan instabil folyamatot eredményezne.

7.2. Átlagtól való távolságot minimalizáló módszer

Az átlagos távolság maximalizálásának talán legnagyobb hátránya, hogy a veszteségfüggvény minden egyes kiszámításakor végig kell iterálnunk az összes eddig létrehozott modell paramétervektorán, amely egy meglehetősen számítás- és memóriaigényes feladat, ami ráadásul minden újabb modell létrehozását követően egyre nehezedik. Ebből adódóan kijelenthetjük, hogy ez a módszer nem skálázható jól nagyobb modellekre, és/vagy nagy modellhalmazokra. Ennek orvoslása végett érdemes megvizsgálnunk a DREAM módszer alábbi, módosított veszteségfüggvényét (DREAM-AGGR):

$$Loss_{m+1} = \rho\left(||\mathbf{\Theta}_m - \theta||_2\right)^{\tau} + \lambda\sum_{j=1}^{N} logp(y_j|x_j,\theta) \tag{7}$$

$$\mathbf{\Theta}_m = \frac{1}{m}\sum_{i=1}^{m}\theta_i \tag{8}$$

Itt a legfontosabb különbség a 7.1. bekezdésben említett formulához képest, hogy itt az eddigi modellektől vett átlagos távolság helyett az eddigi modellek átlagától vett távolságot ($\mathbf{\Theta}_m$) maximalizáljuk a paramétertérben, amelyet modelltanításonként csak egyszer kell kiszámolnunk. Ebből adódóan ez a módszer sokkal jobban skálázható nagyobb modellhalmazokra, ugyanis a loss kiszámítása során mindig egy a tanítás előtt már előre kiszámított értéket veszünk alapul az összes eddigi modellvektor helyett, amelyek száma minden újabb modellel növekszik. Mindemellett azonban fontos, hogy az átlagtól való távolság és az átlagos távolság fogalma közel sem azonos. Ennek szemléltetése végett elég, ha elképzelünk két egymástól d távolságra lévő modellt a paramétertérben: ilyen esetben a harmadik modell a kettő átlagától, vagyis a két modell között húzott képzeletbeli szakasz felezőpontjától maximalizálja a távolságát, ami azt jelenti, hogy a regularizáció mértéke egyforma lesz az említett felezőponttal megegyezű középpontú, d átmérőjű hipergömb felszínének teljes egészén, amely felület magában foglalja a már meglévő két mintát is. Ebből

[3]Pozitív érték esetén ugyanis az eddigi modellektől vett átlagos távolság növekedése növelné a loss-t, és nem csökkentené.

adódóan ezen módszert leginkább akkor érdemes használni, hogyha a 7.1. bekezdésben említett, átlagos távolságot számító módszer alkalmazása a rendelkezésre álló erőforrások szűkössége miatt nem kivitelezhető.

7.3. Összesített távolságot minimalizáló módszer

Végül az előző két bekezdésben említett módszerek mellett érdemes megvizsgálni, annak a lehetőségét, hogy az eddigi modellektől vett távolság helyett a teljes összesített távolságot vesszük figyelembe (tehát az összes eddigi modelltől mért távolságok összegét). Az így módosított veszteségfüggvény (DREAM-SUM) az alábbi:

$$Loss_{m+1} = \rho \left(\sum_{i=1}^{m} ||\theta - \theta_i||_2 \right)^{\tau} + \lambda \sum_{j=1}^{N} logp(y_j|x_j, \theta) \tag{9}$$

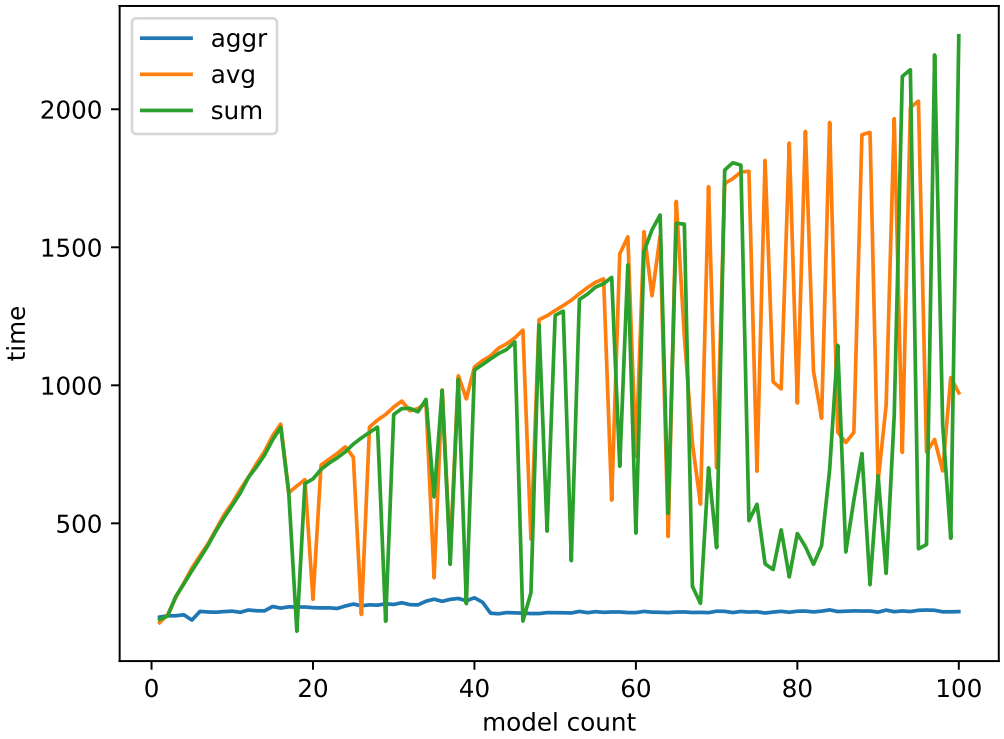
Itt az egyetlen különbség a 7.1 bekezdésben említett módszerhez képest, hogy az összesített távolságnak köszönhetően a loss függvényben kialakuló távolság alapú regularizációs tényező minden újabb modell hozzáadásával monoton módon növekszik, arra kényszerítve ezzel az algoritmust, hogy az eddigi modellektől egyre nagyobb távolságra keressen optimális megoldást a paramétertérben. Habár a DREAM módszer ezen változata kisebb mintahalmaz esetén várhatóan nagyobb diverzitást (vagyis nagyobb átlagos páronkénti távolságot a modellek között) biztosít a másik két változathoz képest, a mintahalmaz növekedésével várhatóan jelentősen romlik az újabb modellek prediktív teljesítménye tekintve, hogy a megnövekedett regularizáció miatt jóval kisebb valószínűséggel találnak megfelelő lokális optimumhelyet a veszteségfüggvényben.

7.4. A DREAM módszer hatékonysága

Ahogyan az a 4. ábrán is látható, a DREAM-AGGR módszer (ábrán: "aggr") a korábbi sejtésünket igazolva konstans tanítási időt produkál, míg a DREAM-AVG és DREAM-SUM módszerek (ábrán: "avg" és "sum") esetén az újabb modellek tanítási ideje a modellek számával lineárisan nő. Az utóbbi két módszer tanítási idejében tapasztalható igen magas variancia annak köszönhető, hogy az egyes modellek tanítása során *checkpoint* és *early stopping* módszereket alkalmaztam, tehát a modellek tanítása automatikusan leállt, hogyha a legutóbbi n darab epoch-ban (esetemben $n = 20$) egy alkalommal sem csökkent a loss értéke az addigi legalacsonyabb érték alá. Mivel ez a feltétel az újabb modellek tanítása során többször is bekövetkezett, így a tanítás nem minden alkalommal ment végig az összesen 200 epoch-on, amely maximumként meg lett állapítva számára, így tehát ezen modellek tanítási ideje is rövidebb volt a vártnál. Ennek ellenére jól látható az említett ábrán a tanítási idő a modellek számának függvényében mért lineáris növekedése.

7.5. Teljesítmény a szintetikus tesztadatokon

Az egyes módszerek a 6.3 bekezdésben bemutatott tesztadatokon mért teljesítménye az 5. ábrán látható.

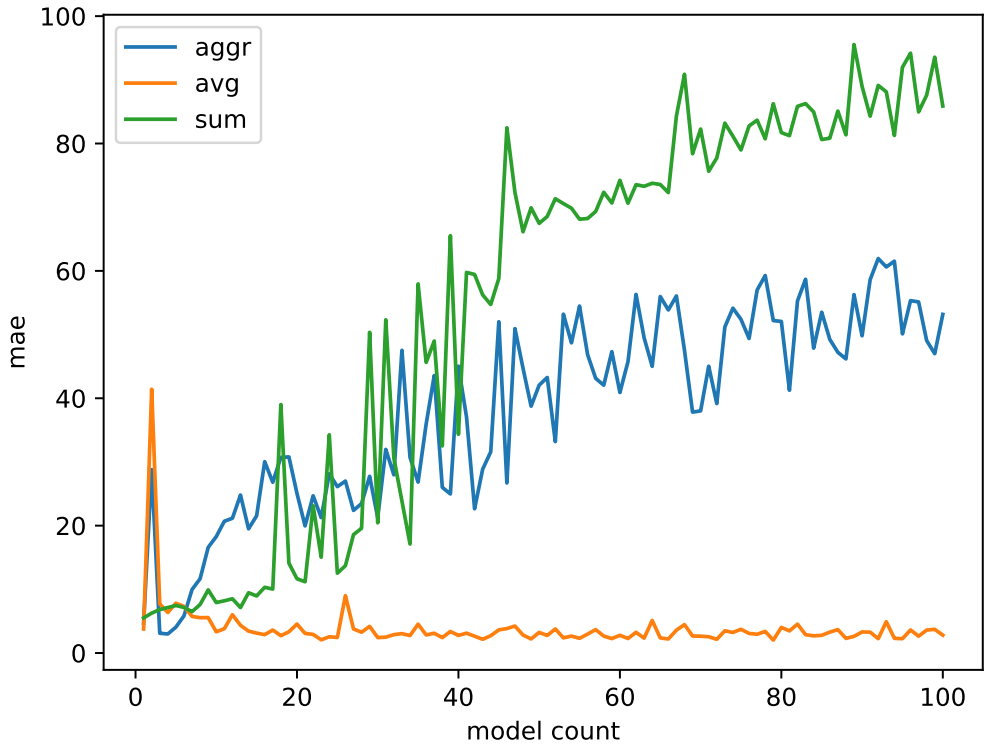


4. ábra. A DREAM módszer változatainak tanítási ideje a modellek számának függvényében (az idő másodpercekben értendő).
("aggr": átlagtól való távolság, "avg": átlagos távolság, "sum": összegzett távolság)

Ahogyan ezen eredményekből is látszik, az itt bemutatott módszerek mindegyikének prediktív teljesítménye jelentősen felülmúlta az SGLD módszert, és a kimenet szórásában is sokkal markánsabb különbség mutatkozik a tanításhoz használt eloszlásból mintavételezett és a véletlenszerűen generált adatok között az említett módszerhez képest. Emellett érdemes megvizsgálnunk, hogy a modellhalmaz mérete mennyiben befolyásolja a prediktív teljesítményt. Az erre vonatkozó eredmények a 6. ábrán láthatók. Ahogyan ezen eredményeken is látszik, egyértelműen az átlagos távolságot minimalizáló (ábrán: avg) módszer produkálta a legjobb és leginkább konzisztens teljesítményértékeket, ami az egyes modellek önálló teljesítményét illeti. Ehhez azonban fontos hozzátenni, hogy a módszerek közötti teljesítménybeli különbség sokkal elenyészőbb, hogyha a teljes modellhalmaz prediktív teljesítményét vesszük figyelembe, ahogyan az a 5. ábrán is látható.

		Training	Validation	Random in range	Random out of range
Mean average error	Mean distance method (avg)	1.4315	2.3909	-	-
	Aggregated mean method (aggr)	4.8531	8.8164	-	-
	Total distance method (sum)	4.9343	8.7254	-	-
Mean standard deviation	Mean distance method (avg)	5.8212	9.5629	49.8154	93.5061
	Aggregated mean method (aggr)	20.2228	36.1957	87.6501	140.7698
	Total distance method (sum)	11.0061	18.7821	59.2042	107.3379

5. ábra. A DREAM módszer három változatának teljesítménye a 6.3 bekezdésben bemutatott tesztadatokon.



6. ábra. A DREAM módszer három változatának ("aggr": átlagtól való távolság, "avg": átlagos távolság, "sum": összegzett távolság) prediktív teljesítménye (*mean average error*) a 6.3 bekezdésben bemutatott tesztadatokon, a modellhalmaz méretének függvényében, mindig a legújabb modellre számolva.

Index	Címke	Osztály neve	Színkód
0	`road`	úttest	(0, 0, 128)
1	`sidewalk`	járda	(0, 74, 55)
2	`building_wall_fence`	épület, fal vagy kerítés	(0, 70, 255)
3	`pole`	oszlop	(0, 224, 255)
4	`traffic_light`	közlekedési lámpa	(0, 250, 176)
5	`traffic_sign`	közlekedési tábla	(6, 254, 20)
6	`vegetation_terrain`	növényzet vagy terep	(103, 212, 0)
7	`sky`	égbolt	(145, 255, 23)
8	`person_rider`	gyalogos vagy sofőr	(218, 255, 31)
9	`car`	személyautó	(255, 227, 0)
10	`truck`	teherautó	(255, 188, 12)
11	`bus`	busz	(255, 66, 0)
12	`train`	vonat	(255, 0, 70)
13	`motorcycle`	motor	(213, 20, 255)
14	`bicycle`	bicikli	(205, 97, 244)
15	`other`	egyéb	(243, 178, 244)

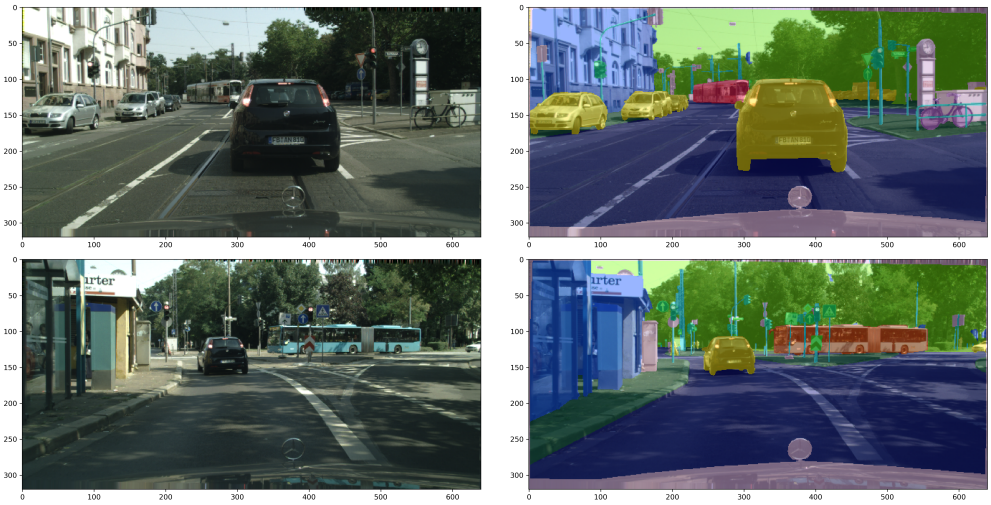
1. táblázat. A CityScapes adathalmazból felhasznált osztályok neve és indexe.

8. A CityScapes adathalmaz

Az eddig ismertetett módszerek gyakorlatban való alkalmazhatóságának vizsgálatához a CityScapes [Cor+16] nevű, főként önvezető autók fejlesztéséhez használt adathalmazt használtam fel. Ezen adathalmaz egy autó szélvédőjére rögzített elülső kamera képeit tartalmazza, amelyek első sorban városi környezetben készültek. Az ehhez kapcsolódó feladat a képeken látható, közlekedés szempontjából releváns objektumok felismerése és osztályozása úgy, hogy a kép minden pixeléről megállapítjuk, hogy melyik képen látható objektumhoz tartozik. Ilyen bemenet-kimenet párból összesen 3500 darabot tartalmaz, amelyből 3000 példa a tanító, 500 pedig a validációs adathalmaz része. Példaként két bemenet-kimenet pár a validációs halmazból a 7. ábrán látható. Az egyszerűség kedvéért az eredetileg az adathalmazban elérhető 33 osztály közül 16-ot használtam fel, az irreleváns, illetve alulreprezentált osztályok elhagyását (tehát az "egyéb" osztályba sorolását) vagy összevonását követően. Ezen objektumosztályok nevei és a hozzájuk tartozó index az 1. táblázatban láthatók.

9. Gyakorlati megvalósítás

Korábban láthattuk, hogy több, meglehetősen különböző megközelítést képviselő módszer is létezik bayesi neurális hálózatok, tehát a paraméterek fölötti a posteriori eloszlás közelítésére. Talán a legfontosabb közös probléma, amely az összes említett módszert érinti, hogy a paraméterek eloszlásának ismeretében történő következtetéshez továbbra is modell-példányokat kell mintavételeznünk ezen eloszlásból, majd ezen példányok kimenetéből következtethetünk a kimenet eloszlására az adott be-



7. ábra. Kettő, a CityScapes adathalmazban található bemeneti kép (adatpont) és a hozzájuk tartozó annotált elvárt kimenet.

menet függvényében. Habár ezen problémára megoldást a Distilled SGLD módszer, annak megbízható működéséhez (tehát a "tanuló" modell megfelelő tanításához) a bemeneti tér közel egészét lefedő véletlen mintavételezésre lenne szükség a "tanító" modellen, amely bonyolultabb problémák esetén (mint amilyen a CityScapes adathalmazban történő objektumdetekció) gyakorlatban nem kivitelezhető. Mivel bonyolultabb problémák esetén továbbra is a modellek mintavételezése a legmegbízhatóbb módszer a következtetésre, így tehát érdemes törekednünk a minták számának és méretének, tehát végső soron a létrejött mintahalmaz paraméterszámának csökkentésére a memória- és számításigény minimalizálása érdekében.

9.1. Értékelési szempontok

Az egyes módszerek kiértékelése során több különböző szempontot kell figyelembe vennünk. Ezek közül az első, és legfontosabb a módszerek által produkált modell prediktív teljesítménye, és hogy mennyire jó becslést adnak a kimenet bizonytalanságára. Mivel a bizonytalanság becslésének jósága közel sem annyira triviális, mint a prediktív teljesítmény, így erre a 9.6. bekezdésben teszek javaslatot. A másik fontos szempont pedig a módszerek erőforrásigénye, amely legfőképp attól függ, hogy a megbízható működésükhöz hány darab minta szükséges, illetve mekkora ezen minták mérete egy adott modell-architektúra mellett. Végső soron tehát egy módszer akkor számít jónak az adott probléma tekintetében, hogyha képes rá egy jó prediktív teljesítményű modellt (illetve modellhalmazt) produkálni, amely jó közelítést ad a kimenet bizonytalanságára, és a paraméterszáma minimális.

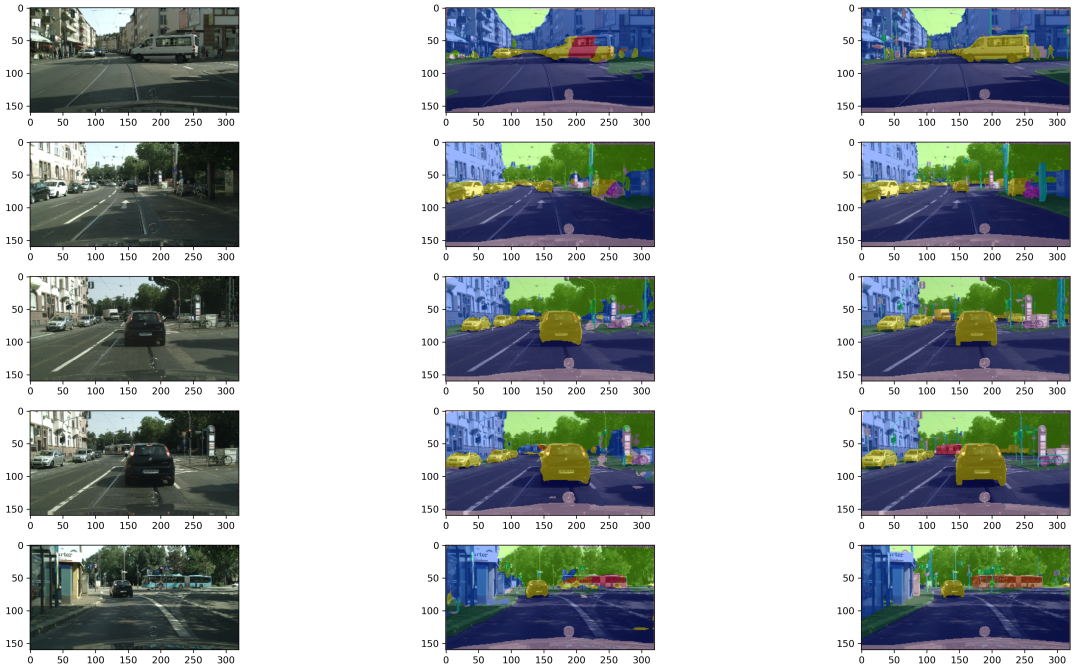
9.2. Az alap modell

Tekintve, hogy az itt ismertetett módszerek mindegyike (és általában a bayesi neurális hálók többsége) lényegében bármilyen létező modell-architektúra fölött alkalmazható, így az egységesség jegyében mindegyiket ugyanazon konvolúciós háló

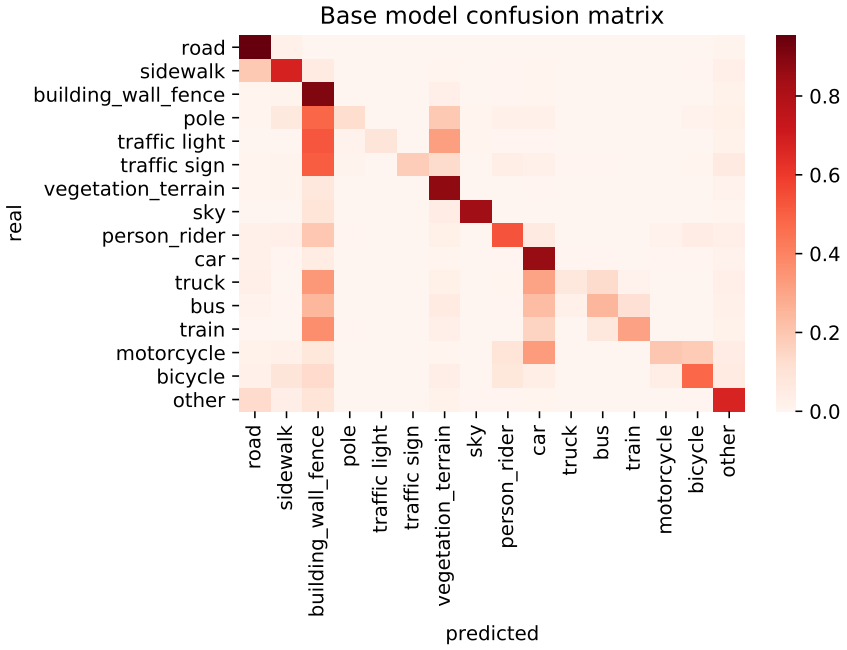
Index	Réteg neve	Kernel	Filterek száma	Kimenet formátuma
0	Input	-	-	(160, 320, 3)
1	Encoding_Conv2D	(3, 3)	32	(160, 320, 32)
2	MaxPooling2D	(2, 2)	-	(80, 160, 32)
3	Encoding_Conv2D	(3, 3)	64	(80, 160, 64)
4	MaxPooling2d	(2, 2)	-	(40, 80, 64)
5	Encoding_Conv2D	(3, 3)	128	(40, 80, 128)
6	MaxPooling2d	(2, 2)	-	(20, 40, 128)
7	Middle_Conv2D	(3, 3)	256	(20, 40, 256)
8	Middle_Conv2D	(3, 3)	256	(20, 40, 256)
9	UpSampling2D	(2, 2)	-	(40, 80, 256)
10	Decoding_Conv2D	(3, 3)	128	(40, 80, 128)
11	UpSampling2D	(2, 2)	-	(80, 160, 128)
12	Decoding_Conv2D	(3, 3)	64	(80, 160, 64)
13	UpSampling2D	(2, 2)	-	(160, 320, 64)
14	Decoding_Conv2D	(3, 3)	32	(160, 320, 32)
15	Output_Conv2D	(3, 3)	16	(160, 320, 16)

2. táblázat. A CityScapes adathalmazon való teszteléshez készített alap modell architektúrája.

architektúrán értékelem ki. Ezen alapmodell rétegeinek felsorolása a 2. táblázaton látható. Ezen modellstruktúrával 84.25%-os pontosságot sikerült elérni a validációs adathalmazon, a modell néhány kimenete pedig a 8. ábrán látható. Ahogyan a kimeneteken is látszik, a modell jól felismeri a gyakoribb osztályokat (mint például úttest, autó, növényzet, stb...), néhány alulreprezentált osztállyal azonban problémába ütközik. Ilyen problémás osztály például a busz, amelyet a példában lévő első és utolsó képen is eltéveszt. Az első képen egy kisbuszt (amely a hozzárendelt címke szerint személyautó) részben busznak, részben személyautónak osztályoz, az utolsó képen pedig egy buszt (amely a címke szerint is busz) vonatként ismer fel. Az ilyen típusú tévedések gyakoriságának felmérése érdekében érdemes megvizsgálnunk a modell konfúziós mátrixát, amely lényegében azt tartalmazza, hogy mely osztályokra milyen gyakorisággal ad egyes predikciókat. Ezen mátrix a 9. ábrán látható. Az itt mutatott eredmények szerint a modell leggyakrabban épületnek osztályozza a nehezen felismerhető, vagy alulreprezentált környezeti objektumokat (például: oszlop, közlekedési lámpa, közlekedési tábla), míg az alulreprezentált járművek többségét személyautónak, amely a leggyakrabban előforduló jármű. Előbbi értelemszerűen jelentős problémát eredményezhet, ugyanis a közlekedési lámpák és táblák felismerése egy önvezető autó szempontjából kritikus. A különböző járművek közötti tévedés (tehát amikor egy járműtípust tévesen másik járműtípusként ismer fel, pl. buszt autónak osztályoz) első sorban az útvonaltervezéssel és potenciális veszélyforrások felismerésével kapcsolatban jelenthet problémát, ugyanis egy teherautónak tipikusan jóval nagyobb a féktávolsága mint egy busznak, a busz pedig nagyobb féktávolsággal rendelkezik, mint egy személyautó. Ebből adódóan az ilyen esetekben létfontosságú felismerni a modell tévedését, tehát futásidőben meghatározni a tévedés valószínűségét a modell bizonytalansága alapján.



8. ábra. Az alap modell kimenetei a validációs adathalmazon. Az oszlopok balról jobbra: bemenet, kimenet, elvárt kimenet.

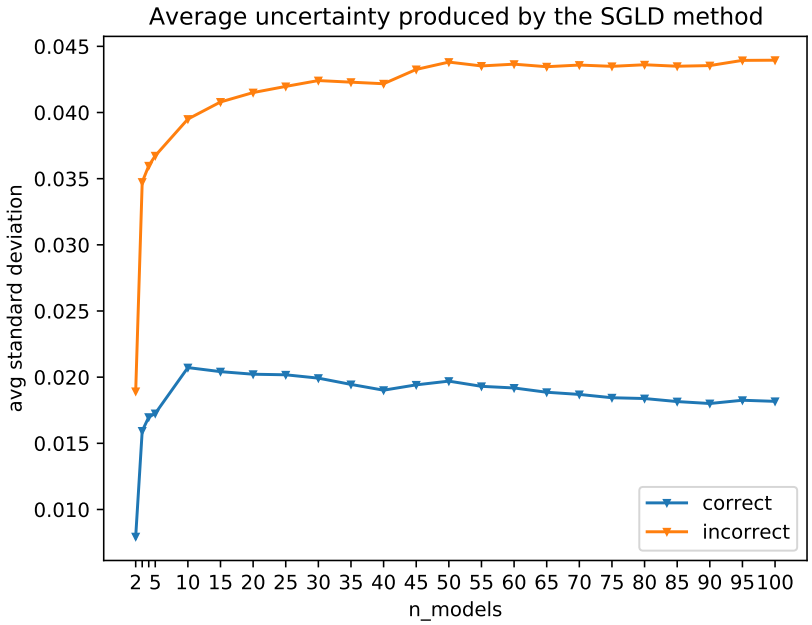


9. ábra. Az alap modell konfúziós mátrixa a validációs adathalmazon.

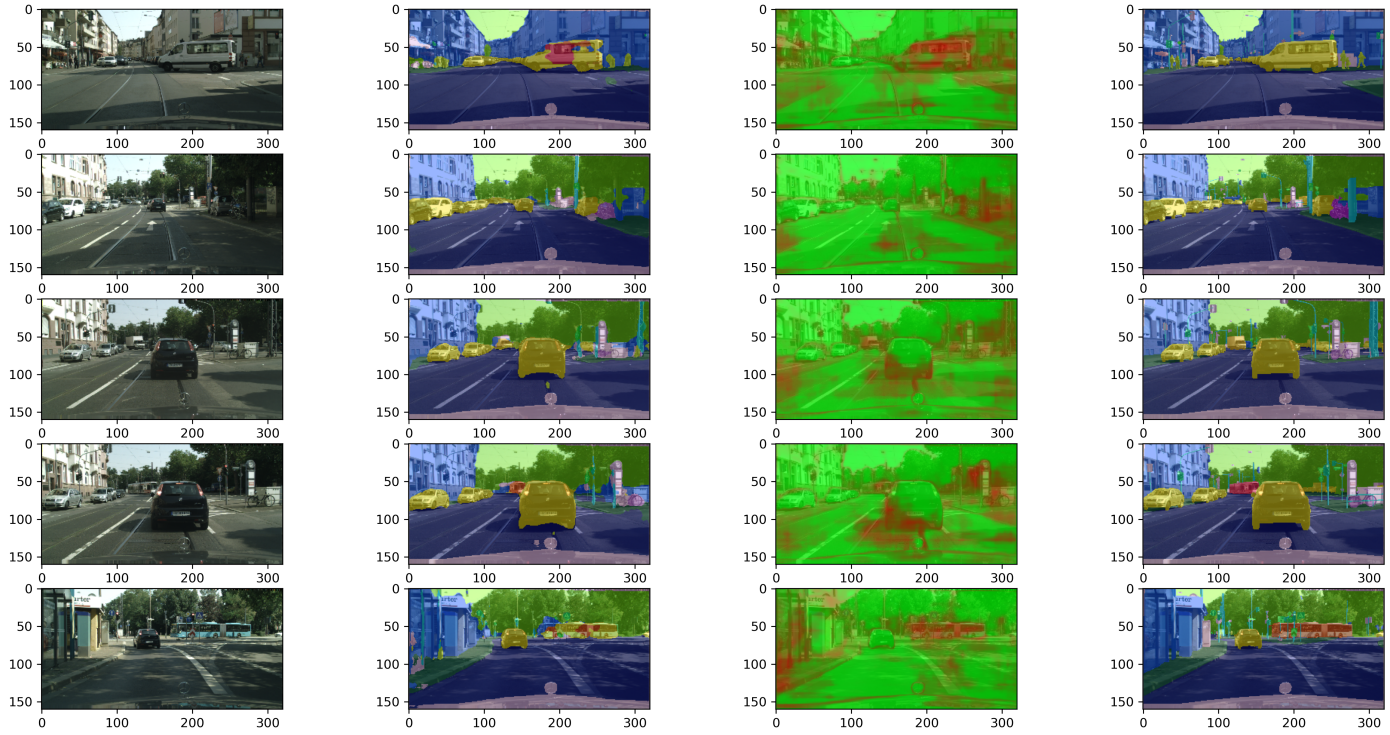
9.3. Az SGLD módszer eredményei

Az SGLD módszer alkalmazásakor az erőforrásigény további csökkentése érdekében csupán a modell középső két rétegének változóit kezeltem valószínűségi változóként, amely rétegek a `Middle_Conv2D` nevet viselik a 2. táblázatban. Ehhez igazodva a tanítás "burn-in" periódusának végén ezen két réteg kivételével az összes többi réteg változóinak értékét (tehát az `Encoding_Conv2D` és `Decoding_Conv2D` rétegek változóit) "befagyasztottam", így tehát a további tanítási lépésekben már csak a középső két réteg vett részt. Ennek első sorban az az előnye, hogy mintavételenként (az első mintavétel kivételével) csak a tanításban részt vevő rétegek változóit kell lementenünk, ugyanis a többi változó értéke fix, így azokat fölösleges többször eltárolni. Emellett további előny, hogy bemenetenként az `Encoding_Conv2D` rétegek kimenetét csak egyszer kell kiszámolni, amellyel további számítási kapacitást spórolhatunk. Végül pedig az eredmények ismertetése előtt még fontos megemlíteni, hogy a modell minden pixelre egy 16 értékű, *one-hot* kódolású kimenetet ad, így tehát a kimenetnek mind a 16 lehetséges kategóriára külön szórás (közvetetten: bizonytalanság) értéket ad. Ezekből többféle képpen állíthatunk elő adott pixelekre vonatkozó bizonytalanságot, én a lehetséges módszerek közül az összes kategóriára adott szórások átlagát vettem alapul. Mindezek fényében a modell kimenete a korábban a 9.2. bekezdésben is látott bemenetekre a 11. ábrán látható. Ahogyan a kimenet is mutatja, ezen modell az alap modellhez hasonló hibákat produkál olyan értelemben, hogy a legelső képen lévő kisbuszt (osztálya szerint személyautót) részben busznak, részben személyautónak osztályozza, az utolsó képen található buszt pedig szinte teljes egészében személyautóként ismeri fel. Fontos azonban, hogy ezen helyeken a bizonytalansága is jelentősen nagyobb (az ábrán: vörösebb), illetve általánosan megfigyelhető, hogy a bizonytalanság leginkább azokon a helyeken magas, ahol a modell ténylegesen téved. Ez tehát azt sugallja, hogy a modell által adott bizonytalanság-értékek valóban jól használhatóak a modell várható tévedésének előrejelzésére. Ennek további vizsgálata végett érdemes megvizsgálnunk a modell konfúziós mátrixa mellett a bizonytalansági mátrixát is, amelynek struktúrája megegyezik a konfúziós mátrixszal, azonban a modell döntései helyett az átlagos bizonytalanságot tartalmazza az összes címke és valós kimenet párra. Ezen két mátrix a 12. ábrán látható. A konfúziós mátrix hasonló értékeket tartalmaz, mint a 9.2. bekezdésben ismertetett alap modell konfúziós mátrixa, így tehát ugyanúgy elmondható erről a modellről is, hogy legfőképp személyautónak vagy épületnek osztályozza az olyan objektumokat, amelyeket nem tud felismerni. A bizonytalansági mátrixon emellett látszik, hogy a helyesen osztályozott (tehát a főátlóban lévő) elemek esetén legtöbb esetben kisebb az átlagos bizonytalanság, mint ott, ahol a modell tévedett. Emellett azon osztályoknak általánosan magasabb a bizonytalansága, amelyek felismerését alulreprezentáltságuk miatt a modell nem tudta rendesen megtanulni (ilyen osztályok például a teherautó, busz, vonat és motorbicikli). Végül pedig érdemes megvizsgálnunk, hogy ezen probléma esetén az SGLD módszerrel mekkora az a legkisebb méretű mintahalmaz, amely már jó becslést ad a modell bizonytalanságára, vagy másképp fogalmazva mennyit nyerünk ezen a téren egy újabb modell hozzáadásával a jelenlegi mintahalmaz méretének függvényében. Ehhez a modell átlagos bizonytalanságát külön a helyes és helytelen predikciók esetén a modellegyüttes méretének függvényében a 10. ábrán mutatom be. Ez alapján látható, hogy a helyes

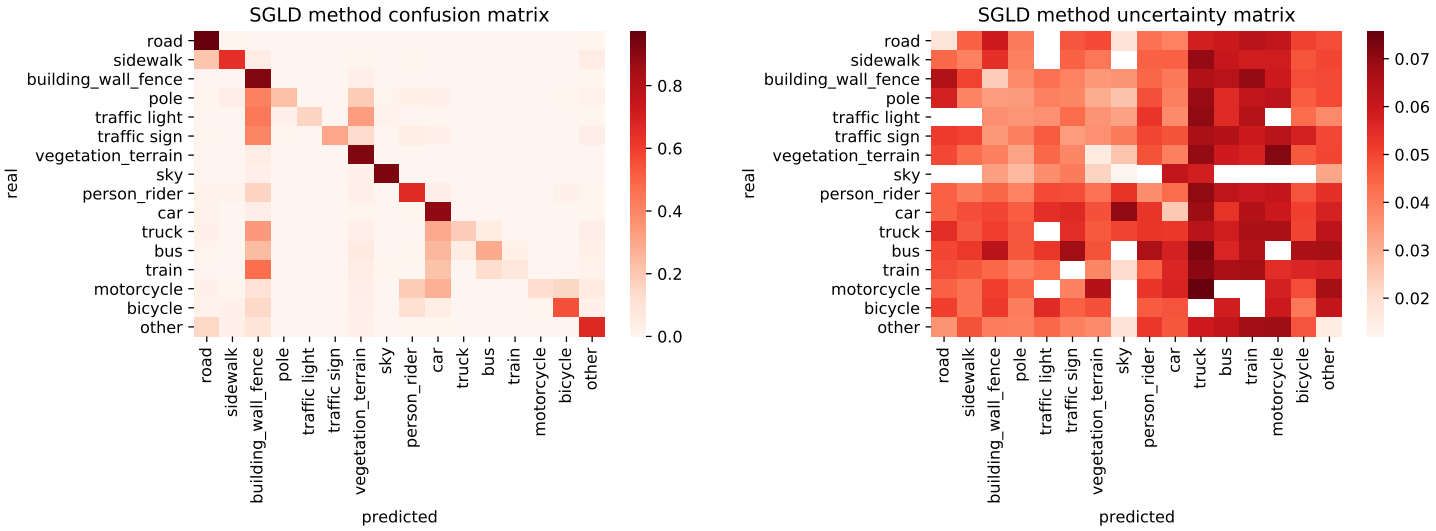
és helytelen válaszokhoz adott átlagos bizonytalanságok közötti távolság a modellek számának növekedésével együtt monoton módon növekszik.



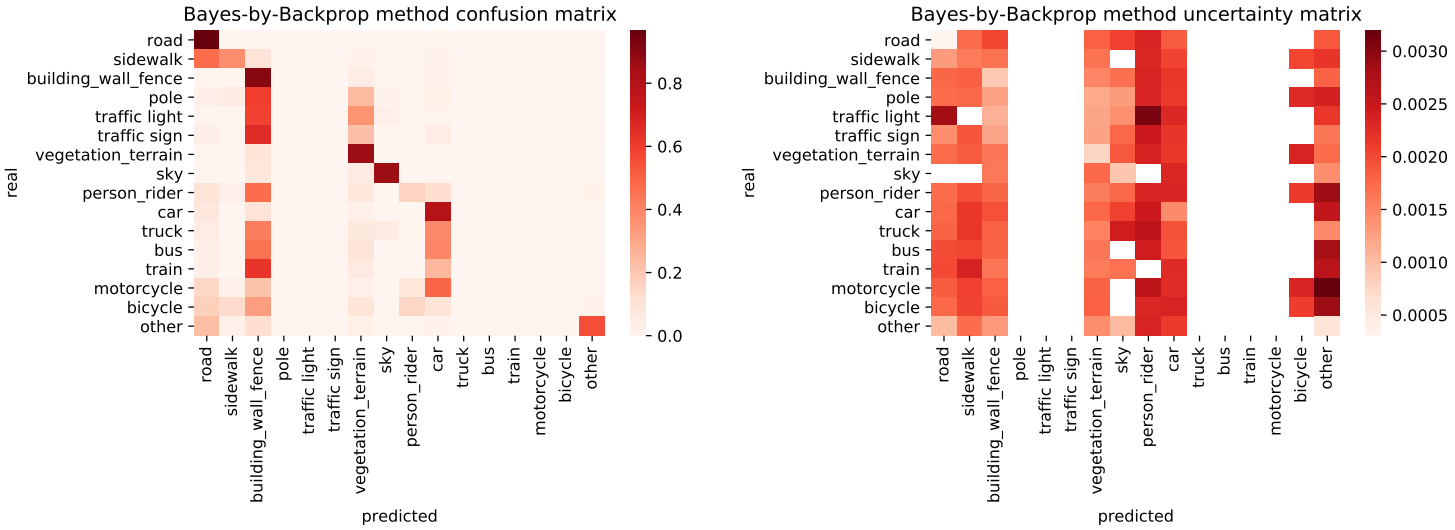
10. ábra. Az SGLD módszer átlagos bizonytalansága helyes és helytelen predikciók mellett a modellegyüttes méretének függvényében.



11. ábra. Az SGLD módszer kimenete a validációs adathalmazon, 25 modellből álló mintahalmaz mellett. Az oszlopok balról jobbra: bemenet, kimenet, bizonytalanság (zöld: alacsony, piros: magas), elvárt kimenet.



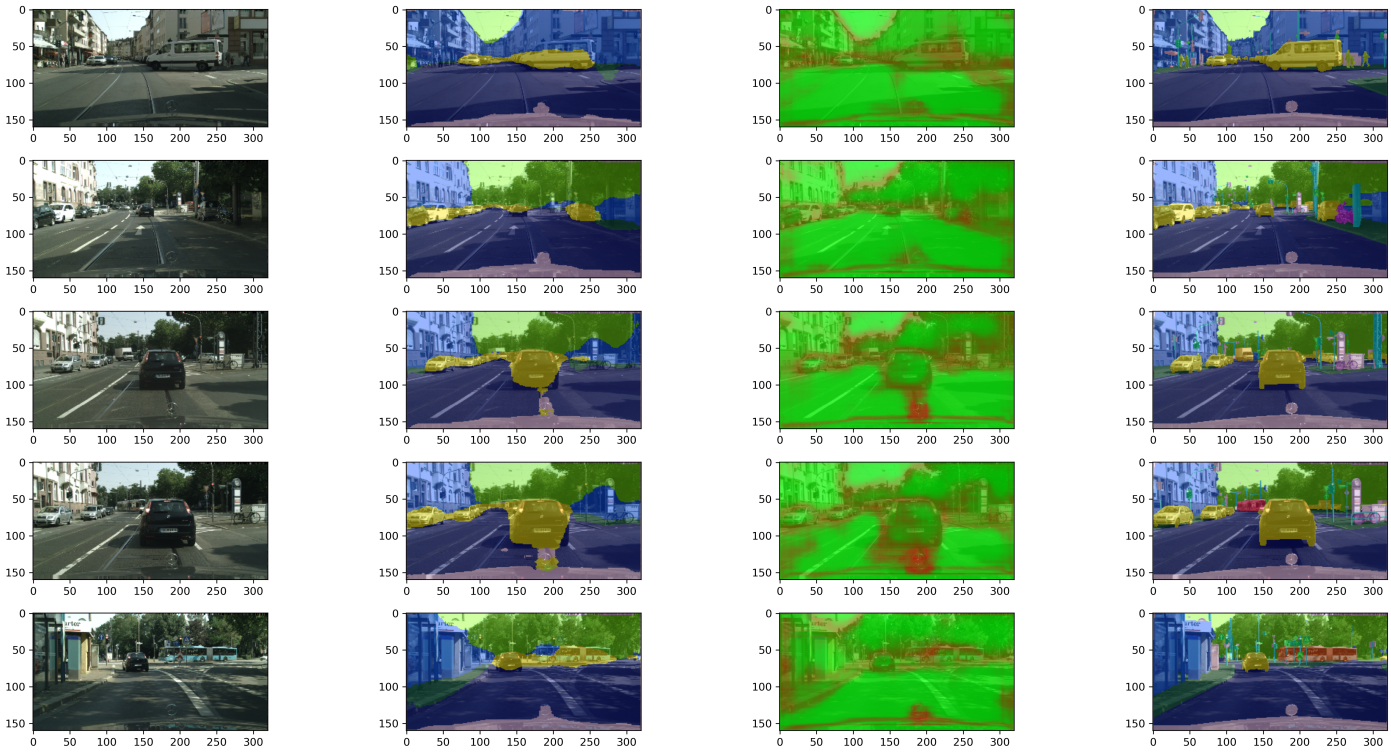
12. ábra. Az SGLD módszer konfúziós mátrixa (balra) és bizonytalansági mátrixa (jobbra) a validációs adathalmazon, 25 modellből álló mintahalmaz mellett.



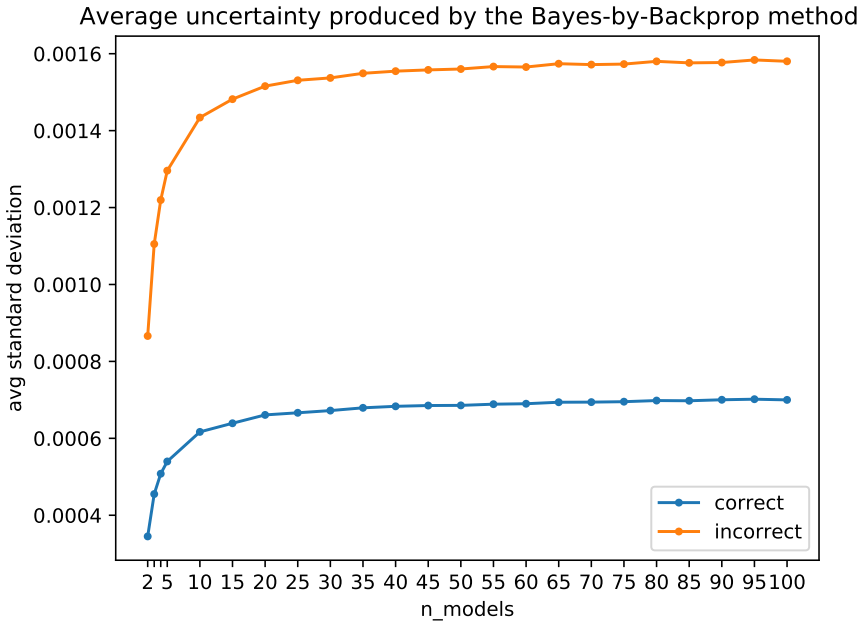
13. ábra. A Bayes-by-Backprop módszer konfúziós mátrixa (balra) és bizonytalansági mátrixa (jobbra) a validációs adathalmazon, 25 modellből álló mintahalmaz mellett.

9.4. A Bayes-by-Backprop módszer eredményei

Az SGLD-hez hasonlóan a Bayes-by-Backprop variációs módszer alkalmazásakor is csupán a középső két réteg (`Middle_Conv2D` rétegek) változói fölött vett eloszlást közelítettem, az erőforrásigény csökkentése érdekében. Így tehát az `Encoding_Conv2D` és `Decoding_Conv2D` rétegek paramétereit normál gradienscsökkentéssel, ezzel párhuzamosan a `Middle_Conv2D` rétegek egymástól független normál a priori eloszlással inicializált változóit pedig a Bayes-by-Backprop algoritmus segítségével tanítottam, előállítva ezáltal az utóbbi rétegek változóinak a posteriori eloszlását úgyszintén egymástól független normál eloszlások formájában. Ezen modell kimenete 25 modellből álló modellegyüttes esetén a 14. ábrán látható, az eddig is használt bemeneti példákra. Ahogyan a kimenetből is sejthető, ezen módszer az adott modell-architektúra mellett sok helyen hibás kimenetet produkál, és a bizonytalansága sem mindig vág egybe a kimenet azon részeivel, ahol téved. Ennek megerősítése végett érdemes megvizsgálnunk a hozzá tartozó konfúziós és bizonytalansági mátrixot, amely a 13. ábrán látható. Ahogyan a konfúziós és bizonytalansági mátrixokból is kiderül, az alulreprezentált osztályok közül a modell szinte egyikre sem adott predikciót. Ennek az oka nagy valószínűséggel a variációs rétegek által bevezetett regularizáció, illetve az ezáltal a modellben kialakuló *bias*. Természetesen nem kizárt, hogy ettől eltérő modell-architektúra esetén a módszer jobb osztályonkénti prediktív teljesítményre is képes. Ennek ellenére tanulságos lehet megvizsgálnunk a modell helyes és helytelen kimenetekhez adott átlagos bizonytalanságát, amely a 15. ábrán látható. Eszerint a helyes és helytelen predikciókra adott átlagos bizonytalanságok közötti távolság a modellegyüttes méretének növekedése mellett a 20-nál nagyobb méretű mintahalmazokra stagnál. Ebből adódóan a jelenlegi problémára (és alap modell-architektúrára) nem érdemes az említettnél nagyobb méretű modellegyüttest létrehozni a Bayes-by-Backprop módszer használata mellett.



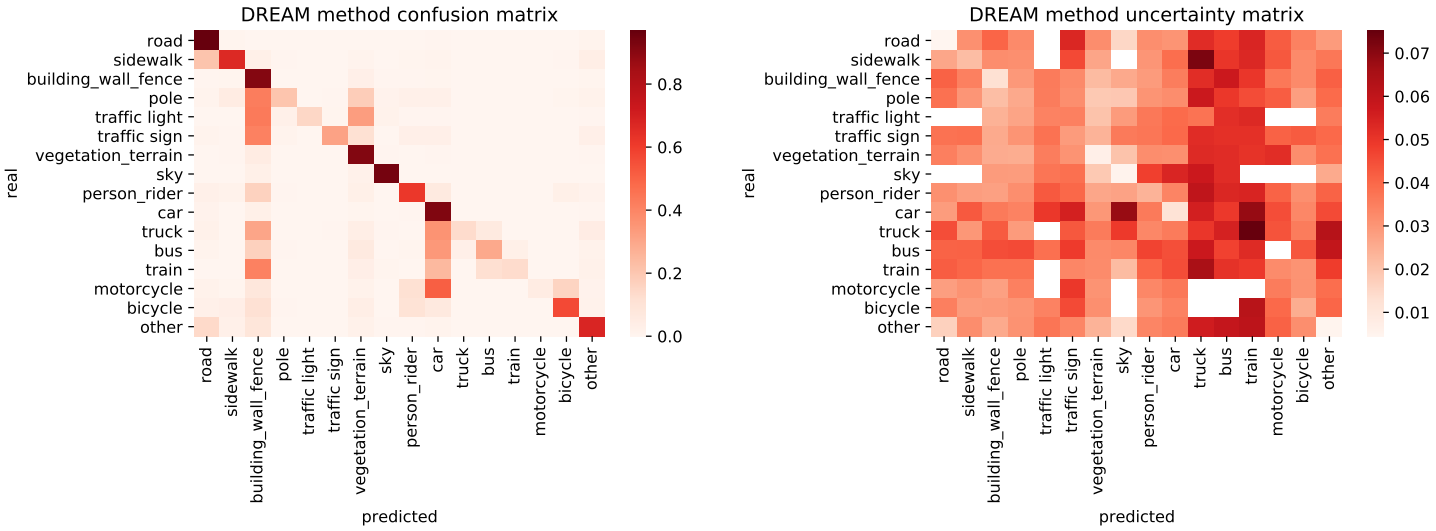
14. ábra. A Bayes-by-Backprop módszer kimenete a validációs adathalmazon, 25 modellből álló mintahalmaz mellett. Az oszlopok balról jobbra: bemenet, kimenet, bizonytalanság (zöld: alacsony, piros: magas), elvárt kimenet.



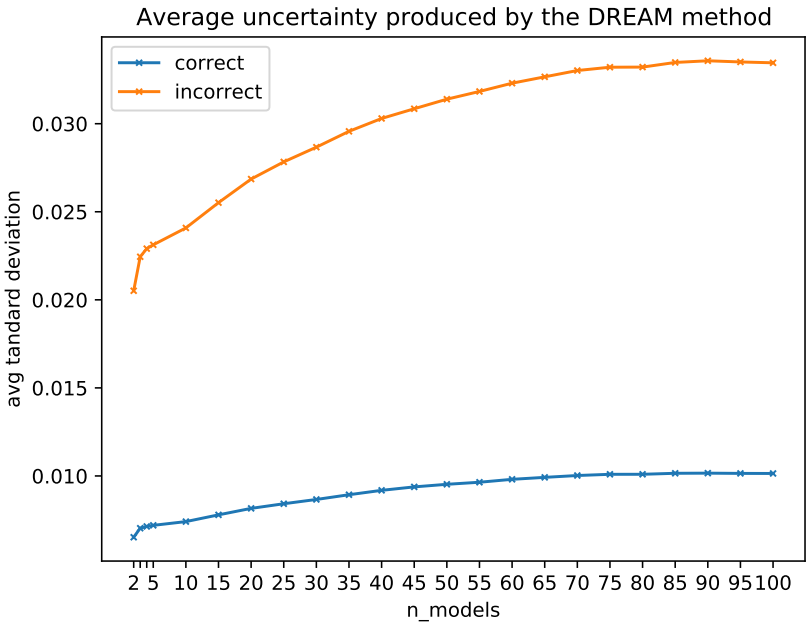
15. ábra. A Bayes-by-Backprop módszer átlagos bizonytalansága helyes és helytelen predikciók mellett a modellegyüttes méretének függvényében.

9.5. A DREAM módszer eredményei

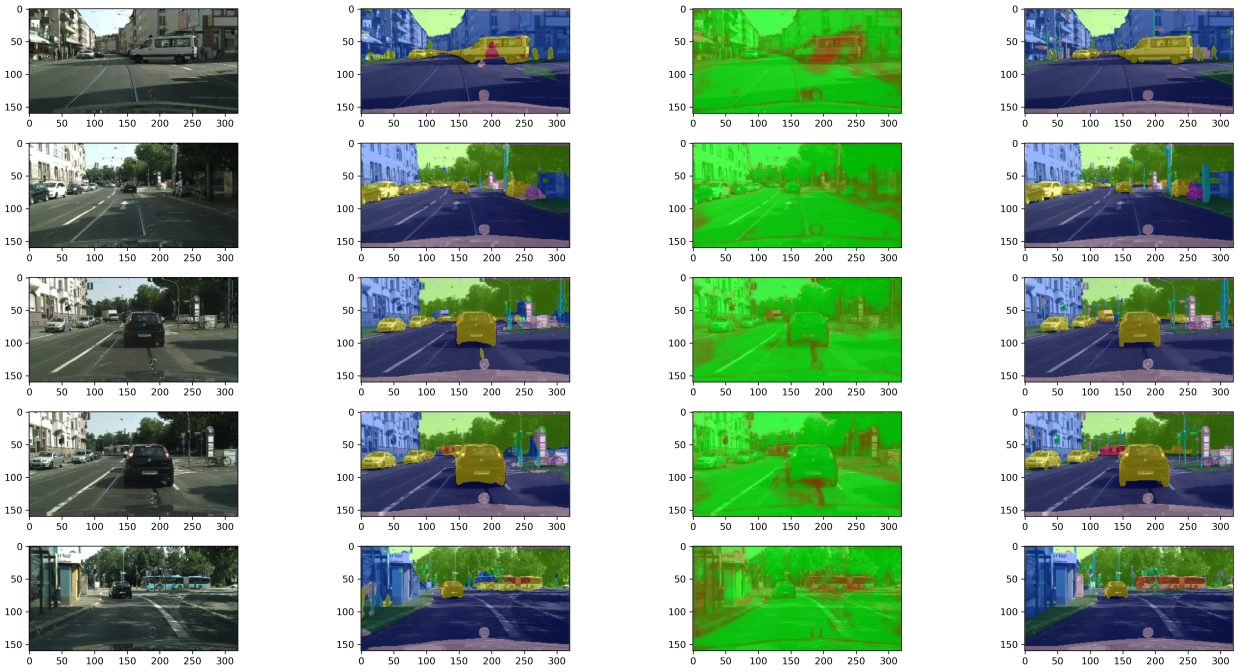
A DREAM különböző változatai közül a DREAM-AVG módszer produkálta a legjobb prediktív teljesítményt a szintetikus teszt adathalmazon a 7.5. bekezdés tanulsága szerint, így ezen változaton keresztül vizsgáltam a módszer gyakorlati teljesítményét a CityScapes adathalmazon. Az erőforrásigény minimalizálása és az egységesség érdekében az eddigi módszerekhez hasonlóan a 9.2. bekezdésben ismertetett architektúrát használtam fel, illetve szintén csak a középső, a 2. táblázatban `Middle_Conv2D` névvel szereplő rétegek változói fölött értelmezett eloszlást közelítettem. Ehhez a modellegyüttes (vagyis mintahalmaz) első tagjaként egy egyszerű, az alap modellhez hasonló modellt tanítottam be, majd az ezt követő újabb modellek tanítása során az `Encoding_Conv2D` és `Decoding_Conv2D` rétegek paramétereinek értékeit az első modellből vettem át, és csak a `Middle_Conv2D` rétegek paramétereit tanítottam, a már meglévő modellektől való átlagos távolságot regularizációként bevezetve. Az így létrejött első 25 modellből álló modellegyüttes kimenetét a 18. ábra mutatja, a már korábban is használt bemenetekre a validációs adathalmazból. Ebből látható, hogy ezen modellegyüttes is többségében ugyanott téved, ahol a 9.3. bekezdésben bemutatott SGLD módszer azonos számú modellből álló együttese, és ugyanúgy magas a bizonytalansága azokon a helyeken, ahol a predikciója hibás. Emellett azonban az említett módszerhez képest láthatóan kevesebb olyan helyen bizonytalan, ahol a predikció nagyrészt helyes. További összehasonlítás végett megvizsgáltam a modellegyüttes konfúziós és bizonytalansági mátrixát, amelyek a 16. ábrán láthatók. A konfúziós mátrix jelentős hasonlóságot mutat az SGLD-nél látotthoz, a bizonytalansági mátrix főátlójában azonban jelentősen alacsonyabb értékek figyelhetők meg az említett módszerhez képest. Végül a DREAM módszer vizsgált változatának helyes és helytelen predikciók mellett adott átlagos bizonytalansága a modellegyüttes méretének függvényében a 17. ábrán látható. Ebből jól látszik, hogy a helyes és helytelen predikciókhoz adott bizonytalanságok közötti távolság monoton módon, de erősen lecsengő mértékben növekszik a modellek számának növekedése mellett.



16. ábra. A DREAM-AVG módszer konfúziós mátrixa (balra) és bizonytalansági mátrixa (jobbra) a validációs adathalmazon, 25 modellből álló mintahalmaz mellett.



17. ábra. A DREAM-AVG módszer átlagos bizonytalansága helyes és helytelen predikciók mellett a modellegyüttes méretének függvényében.



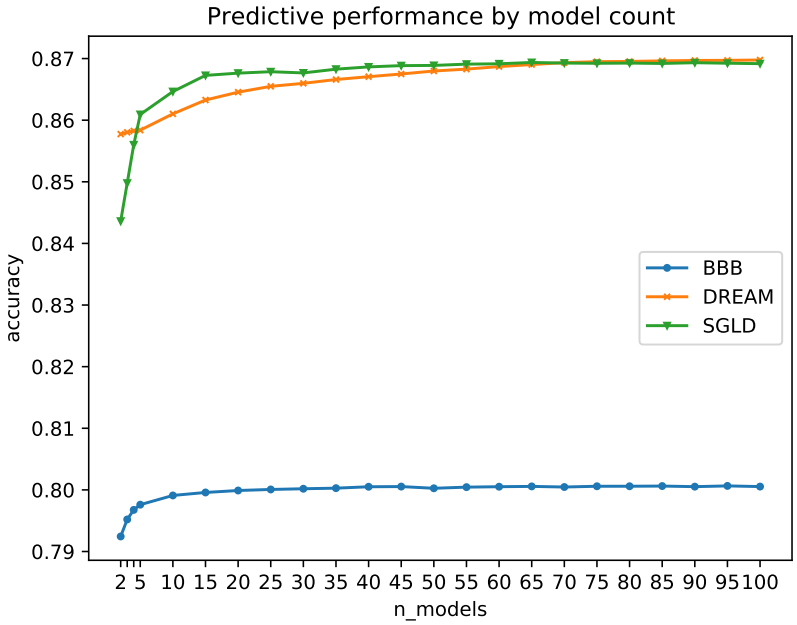
18. ábra. A DREAM-AVG módszer kimenete a validációs adathalmazon, 25 modellből álló mintahalmaz mellett. Az oszlopok balról jobbra: bemenet, kimenet, bizonytalanság (zöld: alacsony, piros: magas), elvárt kimenet.

9.6. A módszerek teljesítményének összevetése

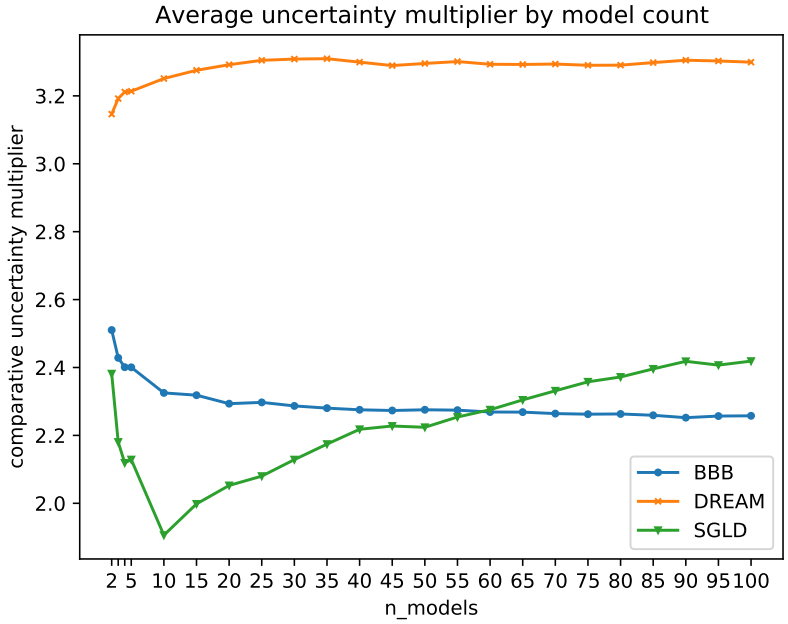
Az első és talán legfontosabb szempont, amely alapján az ismertetett módszereket érdemes összehasonlítani, az a gyakorlati alkalmazásban produkált prediktív teljesítmény. Ezen mérték a 19. ábrán látható mindhárom ismertetett módszerre, a modellegyüttes méretének függvényében. Eszerint az SGLD és DREAM-AVG módszerek hasonlóan jó pontosságú predikciókat adtak a validációs adathalmazon (86.5%), míg a Bayes-by-Backprop módszer prediktív teljesítménye (80%) tőlük jelentősen elmarad, ahogyan az a korábbi kiértékelés alapján is sejthető volt. Emellett megfigyelhető, hogy míg a 10-nél nagyobb mintahalmazokra az SGLD és Bayes-by-Backprop módszerek pontossága stagnál, a DREAM-AVG módszer pontossága habár lecsengő, de monoton növekedést mutat a modellegyüttes méretének növekedésével együtt. Habár ez nem kapott nagy hangsúlyt a korábbi fejezetekben, a 10, 15. és 17. ábrákat megfigyelve észrevehetjük, hogy bár mindegyik módszernél jelentős különbség figyelhető meg a helyes és helytelen kimeneteknél adott átlagos bizonytalanság között, ezen bizonytalanság-értékek skálája módszerenként jelentősen eltér. Ebből adódóan pusztán a két említett átlagos bizonytalanság-érték között vett különbség alapján nem lehet összehasonlítani a módszereket. Ennek megoldása végett vegyük a helytelen és helyes válaszok mellett adott bizonytalanságok hányadosát, amely lényegében azt fejezi ki, hogy hányszor nagyobb az átlagos bizonytalanság abban az esetben, amikor a modell téved, mint akkor amikor a modell által adott predikció helyes. Ezen mérték tehát az alábbi:

$$m(x_n) = \frac{U_{incorrect}(x_n)}{U_{correct}(x_n)} \tag{10}$$

ahol $U_{incorrect}(x_n)$ az x módszer n darab mintából álló modellegyüttesének átlagos bizonytalansága a helytelen kimenetek esetén, $U_{correct}(x_n)$ pedig az x módszer n darab mintából álló modellegyüttesének átlagos bizonytalansága a helyes predikcióknál. Ez a mérték képet adhat arról, hogy a kimenet bizonytalansága ténylegesen mennyire hasznos számunkra, amennyiben azt szeretnénk megbecsülni, hogy a modell mekkora valószínűséggel téved az aktuális bemenetre adott predikció egyes részein. Ezen hányados értéke a három módszer mindegyikére a 20. ábrán látható, a modellegyüttes méretének függvényében. Az $m(\text{BBB}_5) \approx 2.55$, $m(\text{SGLD}_5) \approx 2.15$, $m(\text{DREAM-AVG}_5) \approx 3.2$ értékeket megfigyelve beláthatjuk, hogy alacsony n érték mellett a DREAM-AVG módszer jelentősen nagyobb $m(x_n)$ értéket produkál, míg az ábra tanulsága szerint ($n > 10$) mellett az $m(\text{SGLD}_n)$ értéke jelentősen növekszik n növekedésével, ahogyan az az $m(\text{SGLD}_{90}) \approx 2.4$ értékből is jól látható, amely érték viszont továbbra is elmarad az $m(\text{DREAM-AVG}_{90}) \approx 3.25$-höz képest.



19. ábra. A módszerek prediktív teljesítménye (pontossága) a modellegyüttes (mintahalmaz) méretének függvényében a CityScapes adathalmaz validációs adatpontjain.



20. ábra. A módszerek $m(x_n)$ értéke a modellegyüttes méretének (n) függvényében a CityScapes adathalmaz validációs adatpontjain.

10. Összefoglalás

Az eddigi fejezetekben beláttuk, hogy az itt ismertetett módszerek közül alacsony mintaszám (és így viszonylag alacsony erőforrásigény) mellett a DREAM-AVG által adott bizonytalansági metrika produkálta a legnagyobb különbséget (pontosan a legnagyobb $m(x_n)$ értéket) a helyes és helytelen kimenetek bizonytalansága között a
CityScapes adathalmazon és ugyanazon architektúra felhasználásával, így vélhetően ezen módszer bizonytalansága alapján következtethetünk a legjobban arra, hogy a modell predikciója mikor téves, és mikor nem. Ennek a következtetésnek különös jelentősége lehet (a CityScapes példájából kiindulva) például önvezető autók navigálásakor végzendő kockázatelemzés esetén, ugyanis az eltérő kategóriába eső járművek általában más-más tulajdonságokkal rendelkeznek. Így többek között hogyha a modell személyautónak vagy busznak osztályoz egy - jelentősen nagyobb féktávolsággal rendelkező - teherautót, az könnyedén balesethez is vezethet. Hogyha azonban a modell képes megmondani egy adott járműről, hogy az arra az objektumra adott predikciója bizonytalan, akkor ilyen esetekben feltételezhetjük a rosszabbik eshetőséget (esetünkben személyautó vagy busz helyett teherautót), amellyel lehet túlbecsüljük a veszélyt, azonban következésképpen egy jóval biztonságosabb önvezető rendszert kapunk majd, amely jelentősen kisebb eséllyel becsüli alul egy jármű féktávolságát, és így jóval alacsonyabb valószínűséggel szenved balesetet.

11. Továbbfejlesztési lehetőségek

A gyakorlati megvalósítás során habár sikerült viszonylag jó képet adni a bemutatott módszerek teljesítményéről és hatékonyságáról, az könnyen belátható, hogy ennél sokkal jobb modellek is léteznek a CityScapes adathalmazban történő szemantikus szegmentációra (ilyen modell többek között a HRNet [HRN]). Ebből adódóan érdemes lehet a módszereket az itt ismertetett architektúránál összetettebb modell fölött is kipróbálni, jobb prediktív teljesítmény érdekében. Ezen kívül fontos megemlíteni, hogy variációs következtetésre létezik egy viszonylag újkeletű, és jóval kevésbé elterjedt módszer is, a Stein Variational Gradient Descent [LW19] nevű eljárás. Ezen módszer a szerzők állítása szerint képes a paraméterek fölötti a posteriori eloszlást egymástól nem független formában keresni, amellyel kiküszöböli a Bayes-by-Backprop módszer egyik legnagyobb hátrányát, tekintve, hogy a változók valós a posteriori eloszlásai normál esetben nem függetlenek egymástól. Ebből adódóan erre a módszerre érdemes kiemelt figyelmet fordítani a továbbiakban.

11.1. A szükséges modellek számának további csökkentése

Korábban sok szó esett arról, hogy egy módszer hatékonyságán jelentősen javít, hogyha az általa generált a posteriori eloszlásból elég kevés mintát venni ahhoz, hogy megbízhatóan közelítsük a kimenet eloszlását, illetve maximalizáljuk a helyes és helytelen predikciók bizonytalansága közötti különbséget (pontosabban $m(x_n)$ értéket). Fontos azonban, hogy a megfelelő módszer kiválasztásán kívül mást is

tehetünk a mintahalmaz méretének minimalizálása érdekében. Érdemes megfontolnunk például a már meglévő mintahalmaz egy valódi részhalmazának kiválasztását (valamilyen adott algoritmus segítségével), amelynek tagjai együttesen jó közelítést adnak a kimenet bizonytalanságára jelentősen alacsonyabb modellszám mellett, tovább csökkentve ezzel a modellegyüttes erőforrásigényét a következtetés során.

12. Köszönetnyilvánítás

Szeretném megköszönni a TDK dolgozat és a hozzá kapcsolódó háttérmunka elkészítése során nyújtott segítségét és tanácsadását Dr. Hullám Gábornak, akinek kutatómunkája az Innovációs és Technológiai Minisztérium ÚNKP-20-5 kódszámú Új Nemzeti Kiválóság Programjának a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból finanszírozott szakmai támogatásával és a Bolyai János Kutatási Ösztöndíj segítségével valósult meg.

Hivatkozások

[Blu+15] Charles Blundell és tsai. *Weight Uncertainty in Neural Networks*. 2015. arXiv: 1505.05424 [stat.ML].

[Cor+16] Marius Cordts és tsai. *The Cityscapes Dataset for Semantic Urban Scene Understanding*. 2016. arXiv: 1604.01685 [cs.CV].

[HRN] HRNet. *HRNet Semantic Segmentation*. URL: https://github.com/HRNet/HRNet-Semantic-Segmentation.

[HW20] Eyke Hüllermeier és Willem Waegeman. *Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods*. 2020. arXiv: 1910.09457 [cs.LG].

[Kor+15] Anoop Korattikara és tsai. *Bayesian Dark Knowledge*. 2015. arXiv: 1506.04416 [cs.LG].

[LW19] Qiang Liu és Dilin Wang. *Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm*. 2019. arXiv: 1608.04471 [stat.ML].

[Pea+20] Tim Pearce és tsai. *Uncertainty in Neural Networks: Approximately Bayesian Ensembling*. 2020. arXiv: 1810.05546 [stat.ML].

Táblázatok jegyzéke

Ábrák jegyzéke