# **Dokumentácia**

**Vlastné siete**

Pri vlastných neurónových sieťach sa dáta načítavali z adresárov dát ktoré som rozšíril o dáta upravené v albumentácií. Obrázky som zmenil na rozmer 64x64 pixelov, taktiež som pridal náhodnú zmenu jasu a pridal som zašumenie do obrázkov a na koniec som pridal náhodný posun a rotáciu obrázkov. Dáta boli náhodne rozdelené na trénovacie a testovacie podľa zadania v pomere 80/20 trénovacie a testovacie a neskôr 60/40. Každé trénovanie a testovanie sme opakovali tri krát.

**CNN Sériovo - paralelná**

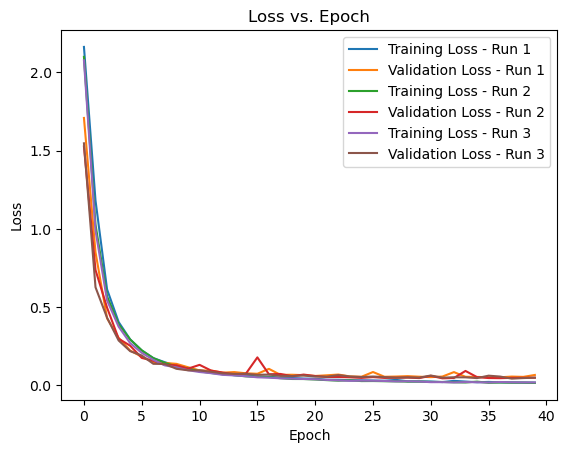
Táto sieť má 3 konvolučné vrstvy s 3 vstupnými kanálmi (na RGB obraz), prvá vrstva má 64 výstupných kanálov a druhá 128, obe vrstvy majú filter o rozmere 5x5 ktorý sa aplikuje po krokoch veľkosti 1 a paddingom veľkosti 2 pixeli. Tretie vrstva má 128 výstupných kanálov a 3x3 maticu s krokovaním a paddingom jedného pixela. Tretia a druhá vrstva bežia paralelne, paralelné vrstvy sa následne vyflattujú a spoja do jednej. Tieto vrstvy sú navrhnuté na zachytenie priestorových prvkov zo vstupného obrazu.

Po každej konvolúčnej vrstve sa výstupné s filtrované obrazy preženú cez Maxpool funkciu ktorá redukuje obraz vybraním dominantných prvkov z obrazu.

Upravený výstup spolu aj s originálnym výstupom prejde site do lineárnych vrstiev kde ku koncu dochádza ku kategorizácií značiek.

V sieti je použitá aktivačná funkcia LeakyRelu a Dropout 20%.

Priemerná úspešnosť siete: 98.77%

****

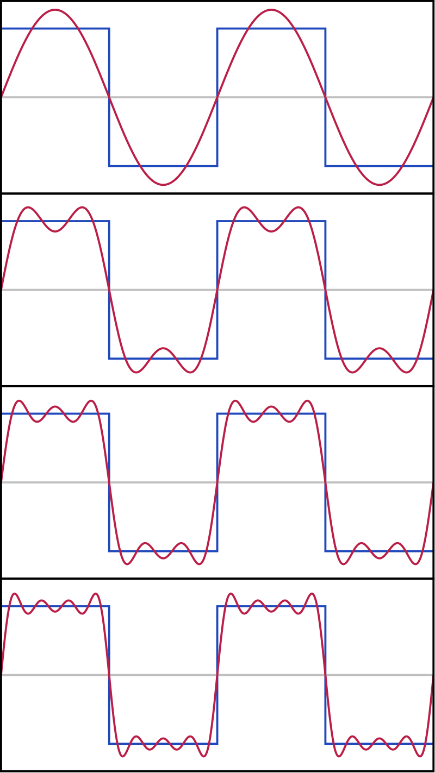
Obr. Graf loss funkcie

**CNN sériová**

Táto sieť má dve konvolučné vrstvy s 3 vstupnými kanálmi (na RGB obraz), prvá vrstva má 64 výstupných kanálov a druhá 128, obe vrstvy majú filter o rozmere 5x5 ktorý sa aplikuje po krokoch veľkosti 1 a paddingom veľkosti 2 pixeli. Tieto vrstvy sú navrhnuté na zachytenie priestorových prvkov zo vstupného obrazu .

Po každej konvolúčnej vrstve sa výstupné s filtrované obrazy preženú cez Maxpool funkciu ktorá redukuje obraz vybraním dominantných prvkov z obrazu.

Následne dôjde k flattovaniu výstupov kde výstupné prvky dosadím do trigonometrických funkcií ktoré sa často používajú vo Fourierových rádov, tieto zmeny majú podobný efekt ako aktivačné funkcie ktoré nám pomáhajú riešiť komplikovanejšie problémy. Fourierové rády sa používajú ako ďalší spôsob pri hľadaní funkcie ako napríklad rôzne formy regresie. Ako vidno na obrázku:



Obr. Príklad Fourierovho rádu

Upravený výstup spolu aj s originálnym výstupom prejde site do lineárnych vrstiev kde ku koncu dochádza ku kategorizácií značiek.

V sieti je použitá aktivačná funkcia LeakyRelu a Dropout 20%.

Priemerná úspešnosť siete: 99.04%

**Obrázok, na ktorom je text, snímka obrazovky, diagram, rad

Automaticky generovaný popis**

Obr. Graf loss funkcie

Následne vidíme confusion matrix pre našu sieť ktorá bola zobraná s testovacích dát.

**Obrázok, na ktorom je text, snímka obrazovky, diagram, číslo

Automaticky generovaný popis**

Obr. Confusion matrix

Ďalej vidíme obrázky a filtre použité v prvej a druhej konvolučnej vrstve.

**Obrázok, na ktorom je vzor, štvorec, pestrofarebnosť, symetria

Automaticky generovaný popis**

Obr. Obrazy s prvej conv. vrstvy

**Obrázok, na ktorom je snímka obrazovky, pestrofarebnosť, štvorec

Automaticky generovaný popis**

Obr. Obrazy s druhej conv. vrstvy

**Obrázok, na ktorom je vzor, štvorec, krížovka, látka

Automaticky generovaný popis**

Obr. filtre s prvej conv. vrstvy

**Obrázok, na ktorom je text, vzor, symetria, čierno-biela

Automaticky generovaný popis**

Obr. filtre s druhej conv. vrstvy

**Pred trénované siete**

Pri všetkých pred trénovaných neurónových sieťach sa dáta načítavali rovnakým spôsobom. Kde koreňový adresár pre trénovacie aj testovacie dáta obsahoval pod priečinky ktoré označovali jednotlivé tipy dát. Dáta boli náhodne rozdelené na trénovacie a testovacie podľa zadania v pomere 80% trénovacie a 20% testovacie. Po načítaní obrázkov bola aplikovaná argumentácia a normalizácia. Následne boli dáta roztriedené a boli im priradené štítky pomocou ktorých budú klasifikované v neurónovej sieti

**Densenet**Po načítaní pred trénovaného modelu sme zmrazili všetky jeho vrstvy a následne vykonali jemné ladenie rozmrazením poslednej plne prepojenej klasifikačnej vrstvy. Tuto vrstvu sme nahradili novou vrstvou ktorá zodpovedá počtu typov vstupných obrázkov. Okrem toho sa nastavuje aj optimalizácia pre tréning modelu.

* V trénovacej slučke máme nastavených 20 epoch kde každá slučka bola vykonaná 3 krát. Valdačné straty sú vyobrazené v obrázku nižšie:
* Obrázky z validácie
* Nakoniec sa úspešnosť modelu vyhodnotila na testovacích dátach.
* Priemerná úspešnosť pri modely zo zmrazenými vrstvami: 82.92%

Obrázok, na ktorom je text, snímka obrazovky, vývoj, diagram

Automaticky generovaný popis

Obr. Graf Loss-function DenseNet

Tretí beh dosiahol úspešnosť: 82.38%

**Alexnet**Po načítaní pred trénovaného modelu sme zmrazili všetky jeho vrstvy a následne vykonali jemné ladenie rozmrazením poslednej plne prepojenej klasifikačnej vrstvy. Tuto vrstvu sme nahradili novou vrstvou ktorá zodpovedá počtu typov vstupných obrázkov. Okrem toho sa nastavuje aj optimalizácia pre tréning modelu.

* V trénovacej slučke máme nastavených 20 epoch kde každá slučka bola vykonaná 3 krát. Validačné straty sú vyobrazené v obrázku nižšie:
* Nakoniec sa úspešnosť modelu vyhodnotila na testovacích dátach.
* Priemerná úspešnosť pri modely zo zmrazenými vrstvami: 87.82%

Obrázok, na ktorom je text, snímka obrazovky, vývoj, diagram

Automaticky generovaný popis

Obr. Graf Loss-function AlexNet

Obrázok, na ktorom je text, snímka obrazovky, číslo, diagram

Automaticky generovaný popis

Obr. Confusion matrix AlexNet

Tretí beh dosiahol úspešnosť: 89.53%

**Resnet18**Po načítaní pred trénovaného modelu sme zmrazili všetky jeho vrstvy a následne vykonali jemné ladenie rozmrazením poslednej plne prepojenej klasifikačnej vrstvy. Tuto vrstvu sme nahradili novou vrstvou ktorá zodpovedá počtu typov vstupných obrázkov. Okrem toho sa nastavuje aj optimalizácia pre tréning modelu.

* V trénovacej slučke máme nastavených 20 epoch kde každá slučka bola vykonaná 3 krát. Validačné straty sú vyobrazené v obrázku nižšie:
* Obrázky z validácie
* Nakoniec sa úspešnosť modelu vyhodnotila na testovacích dátach.
* Priemerná úspešnosť pri modely zo zmrazenými vrstvami: 80.32%

Obrázok, na ktorom je text, snímka obrazovky, vývoj, rad

Automaticky generovaný popis

Obr. Graf Loss-function ResNet18

Tretí beh dosiahol úspešnosť: 80.32%

**Mobilenet**Po načítaní pred trénovaného modelu sme zmrazili všetky jeho vrstvy a následne vykonali jemné ladenie rozmrazením poslednej plne prepojenej klasifikačnej vrstvy. Tuto vrstvu sme nahradili novou vrstvou ktorá zodpovedá počtu typov vstupných obrázkov. Okrem toho sa nastavuje aj optimalizácia pre tréning modelu.

* V trénovacej slučke máme nastavených 20 epoch kde každá slučka bola vykonaná 3 krát. Validačné straty sú vyobrazené v obrázku nižšie:
* Obrázky z validácie
* Nakoniec sa úspešnosť modelu vyhodnotila na testovacích dátach.
* Priemerná úspešnosť pri modely zo zmrazenými vrstvami: 73.35%

Obrázok, na ktorom je text, diagram, vývoj, snímka obrazovky

Automaticky generovaný popis

Obr. Graf Loss-function MobileNet

Tretí beh dosiahol úspešnosť: 74.25%

**Záver**

Z výslodkov presnosti a validačných strát je vidieť že predtrénované siete nedokázali s takou úspešnosťou priradiť jednotlivé obrázky do správnych kategórií. Toto je spôsobené tým že tieto predtrénované siete boli trénované na obrázkoch s väčším rozlíšením zatiaľ čo vstupný dataset používaný pri testovaní obsahoval menšie obrázky z nižším rozlíšením. V rámci experimentu pri opätovnom trénovaní predtrénovaných sietí sa dokázali siete na trénovať a dosiahli pomerne lepšie výsledky. Stále však nie až také dobré ako vlastné konvolučné neurónvé siete.