Výstup 2 maturitní práce

Zařízení na snímání gest

autor:
Štěpán Bílek

konzultant:
Jaroslav Kořínek

Školní rok 2025/2026

1 Úvod

Primárním cílem výstupu 2 maturitní práce bylo seznámit se s principem neuronových sítí a navrhnout jednu pro ovládací program za účelem rozpoznání jednotlivých gest. Cílem sekundárním bylo začít na ovládacím programu samotném, aby byl schopen přijímat data ze zařízení a ukládat je jako trénovací soubory pro neuronovou síť.

2 Neuronová síť

2.1 Poznatky

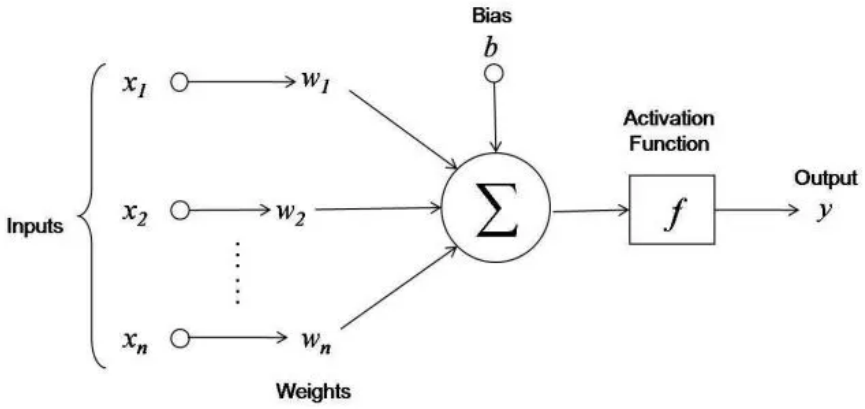
Umělá neuronová síť je struktura určená pro distribuované paralelní zpracování dat. Skládá se z umělých neuronů, jejichž volným předobrazem je biologický neuron. Neurony jsou vzájemně propojeny synaptickými vazbami a navzájem si předávají signály a transformují je pomocí aktivačních funkcí. [1]

Neurony jsou uspořádávány do vrstev. Existuje vrstva vstupní, skrytá (kterých může být více) a výstupní.

2.1.1 Neurony, synapse a vrstvy

Neuron je funkcí, která má libovolný počet vstupů. Tato funkce je ve své podstatě jednoduchá, přijímané vstupy se nejdříve vynásobí vahou vázanou k synapsy, ze které přišly, dále se tyto hodnoty sečtou a nakonec je přičten tzv. „bias“.

Synapse je propojení mezi neurony. Ke každé jednotlivé synapsy, kterou neuron má, je přiřazena váha, která začíná na náhodném čísle a v procesu učení je upřesňována.



Obrázek 1: neuron

Za předpokladu neuronu s 1 vstupem[x], vahou na synapsy[w], bias[b] a jedním výstupem[y] vypadá jeho oprace po matematické stránce takto:

$$y = x * w + b \tag{1}$$

pozn. Jinými slovy se jedná o přímku / lineární funkci.

Počet vrstev a neuronů v každé z nich se stanovuje podle komplexity problému, jež má síť za úkol řešit, a velikosti datasetu. Neexistuje obecné pravidlo pro stanovení ideálního počtu vrtev ani neuronů. Důležité je při trénování modelu sledovat přesnost a podle toho síť adaptovat. Síť musí být dostatečně složitá, aby dokázala produkovat správné výsledky, avšak ne tolik složitá aby si data pouze pamatovala.

2.1.2 Aktivační funkce

Je další funkcí v pořadí za neuronem. Tato funkce není nezbytná pro neuronovou síť, ale zpravidla dělá výstup nelineárním, díky čemuž se síť může naučit komplexnější vzory. Nejčastěji používanými jsou: **sigmoid**, **tanh**, **ReLU** a tomu podobné, **softmax**. Pro určité úkoly na různých vrstvách sítě jsou některé funkce lepší, než jiné. Odvíjí se to od faktorů jako příkrost vzrůstu a H_f.[2]

2.1.3 Ztrátová funkce

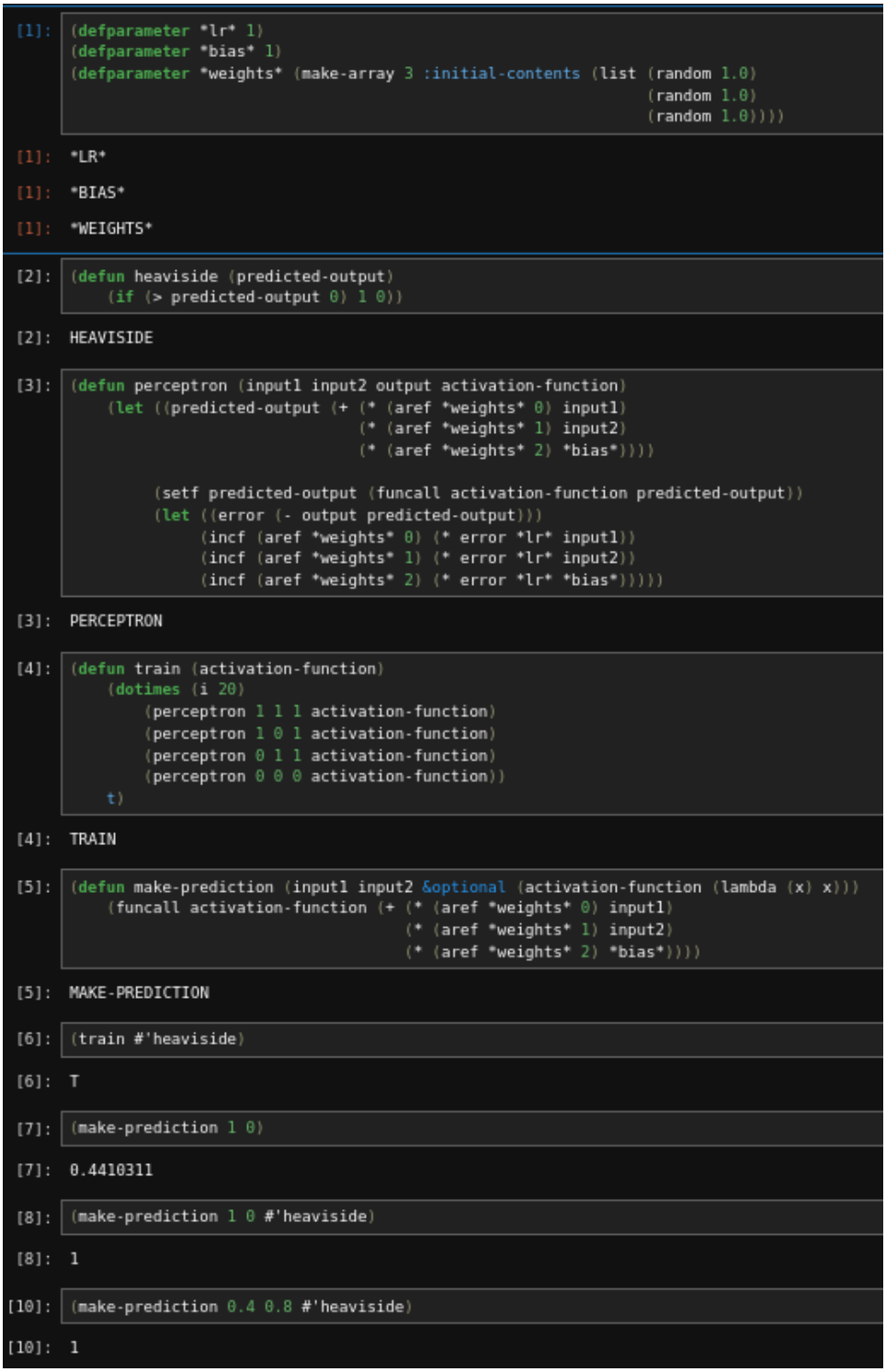
Ztrátová funkce určuje, jak správně model předpovídá/určuje v porovnání s opravdovým výsledkem. Čím menší číslo je výstup této funkce, tím přesnější model je. Tato funkce je důležitá při učení, kdy právě podle chybovosti algoritmy upravují parametry modelu(váhy, bias). Těchto funcí existuje mnoho a každá je lepší na něco jiného[3].

2.1.4 Optimalizační funkce

Optimalizační funkce úzce souvisí se ztrátovou funkcí a na základě jejího výstupu upravuje parametry modelu, jak již bylo výše zmíněno[4].

2.1.5 Implementace perceptronu

Pro lepší pochopení celého tématu jsem implementoval jeden z nejjednoduších modelů neuronové sítě - Perceptron[5].



Obrázek 2: implementace perceptronu

2.2 Návrh

2.2.1 Vrstvy a neurony

Neuronová síť bude sestávat ze 4 vrstev, čili bude obsahovat 2 skryté vrstvy.

Vstupní vrstva bude mít 180 vstupů, jelikož data z gyroskopu a akcelerometru jsou složena z 6 hodnot, délku gesta budu předběžne počítat 2s a frekvenci sběru 15Hz.

První skrytá vrstva bude LSTM(Long Short-Term Memory)[6], která se používá především pro zpracování sekvenčních dat - zachcené hodnoty z gyroskopu a akcelerometru za určitý čas. Měla by dokázat dobře zachtit a rozpoznat průběh gest. Počet neuronů jsem stanovil na 32.

Druhá je vrsva tzv. „hustá“, její charakteristikou je, že všechny neurony jsou připojeny ke všem neuronům z vrtvy předchozí[7]. Počet neuronů bude 16.

Poslední vrstva bude také hustá a bude mít 2 neuron, jelikož nás budou zajímat 2 gesta (doleva, doprava).

pozn. Počet neuronů v jednotlivých vrstvách se pravděpodobně bude v budoucnu během testování sítě měnit, uvedená čísla jsou počáteční.

2.2.2 Aktivační funkce

U první vrstvy bude použit hyperbolický tangens, protože ho LSTM vrstva již vnitřně používá a s jinou aktivační funkcí by vrstva nemusela produkovat správné výsledky.

U druhé skryté vrstevy bude použita funkce **leaky ReLU**, jelikož je jednoduchá na výpočet - což bude hrát roli především u tréninku - a oproti standartní ReLU nemůže dojít k „zaseknutí“ výstupu neuronu na 0.

Na vrstvě výstupní bude použita funkce softmax, která je běžně využívána jedná-li se o klasifikaci výstupu do více tříd. Funkce každé třídě přiřadí pravděpodobnost.

2.2.3 Ztrátová funkce

Ztrátová funce bude „Categorical Cross-Entropy Loss“, protože slouží pro modely, které klasifikují více tříd, jejichž výstupem je pravděpodobnost.

2.2.4 Optimalizační funkce

Jako optimalizátor bude použit alogoritmus „Adam“, především kvůli jeho univezálnosti, dále také nepotřebuje podrobně ladit rychlost učení. Pro síť a dataset této velikosti by měl být ideální.

3 Ovládací program

Ovládací program pro zařízení na snímání gest má za úkol přijímat a vyhodnocovat data ze zařízení na počítači uživatele a dle toho počítač ovládat(posouvat snímky prezentace).

Program je psaný v jazyce Python, kvůli jeho rozsáhlému ekosystému knihoven. Zároveň je Python interpretovaným jazykem a tudíž ho lze spustit na jakémkoli operačním systému, kde běží Python interpreter, bez větších rozdílů. Pro správu virtuálního prostředí, kde jsou knihovny nainstalovány je použit software miniconda. Projekt má strukturu dle standartů a je psaný objektově orientovaným způsobem.

Aktuálně program skoro umí přijímat data ze zařízení a ukládat je jako trénovací pro budoucí neuronovou síť. Aktuální stav má několik problémů, jimiž se zabývá spolupracovník Lukáš Karásek.

Odkazy

[1] URL: https://cs.wikipedia.org/wiki/Um%C4%9Bl%C3%A1_neuronov%C3%A1_s%C3%AD%C5%A5.

[2] URL: https://www.geeksforgeeks.org/machine-learning/activation-functions-neural-networks/.

[3] URL: https://www.geeksforgeeks.org/deep-learning/loss-functions-in-deep-learning/.

[4] URL: https://www.geeksforgeeks.org/deep-learning/optimization-rule-in-deep-neural-networks/.

[5] URL: https://cs.wikipedia.org/wiki/Perceptron.

[6] URL: https://www.geeksforgeeks.org/deep-learning/deep-learning-introduction-to-long-short-term-memory/.

[7] URL: https://www.geeksforgeeks.org/deep-learning/dense-layer-tf-keras-layers-dense-in-tensorflow/.

Obsah