

# Vyučovací software na vizualizáciu učenie neurónových sietí

## *Katalóg Požiadaviek*

Peter Hozlár, Matej Miškovčík, Miroslav Klieštík, Jakub Kováč

## OBSAH

<b>1. Úvod.....</b>	<b>1</b>
1.1 Účel dokumentu.....	1
1.2 Rozsah využitia Systému.....	2
1.3 Slovník pojmov.....	2
1.4 Referencie.....	2
1.5 Prehľad nasledujúcich kapitol.....	2
<b>2. Všeobecný popis.....</b>	<b>3</b>
2.5 Predpoklady a závislosti.....	4
<b>3. Špecifikácia požiadaviek.....</b>	<b>5</b>
3.1 Základné definície.....	5
3.2 Typy ovládacích panelov.....	5
3.3 Funkcionalita hlavného ovládacieho panela.....	5
3.4 Funkcionalita vedľajšieho ovládacieho panela.....	5
3.5 Funkcionalita Renderera.....	6
<b>4. Prílohy.....</b>	<b>7</b>
4.1 Červeno-zelená mapa.....	7
4.2 Trojzložkové rozpoznávanie farieb.....	7

# 1. Úvod

## 1.1 Účel dokumentu

Účelom tohto dokumentu je načrtnúť funkčné požiadavky na vývoj vzdelávacieho softvéru. Tento softvér bude vizualizovať, ako sa neurónové siete učia pomocou algoritmu spätného šírenia. Tento softvér je určený na pomoc pri interaktívnom vyučovaní konceptov neurónových sietí, spätného šírenia a tréningových procesov. Tento dokument je záväzná dohoda medzi zadávateľom a vývojovým tímom.

## 1.2 Rozsah využitia Systému

Softvér poskytuje praktické vzdelávacie prostredie, kde si používatelia môžu vybrať z preddefinovaných neurónových sietí, vizualizovať štruktúru a manipulovať so vstupnými parametrami. Tento softvér bude výstupom projektu v rámci predmetu Tvorba informacných systémov na FMFI UK BA v akademickom roku 2024/2025.

## 1.3 Slovník pojmov

- **Perceptron:** Základná jednotka neurónovej siete, ktorá sa inšpiruje biologickým neurónom.
- **Neurónová sieť (NN):** Systém vzájomne prepojených umelých perceptronov používaných na spracovanie a vyhodnocovanie informácií.
- **Spätné šírenie:** Algoritmus učenia, ktorý spätne upravuje váhy v neurónovej sieti tak že porovnáva aktuálny výstup neurónovej siete s požadovaným cieľom.
- **Aktivačná funkcia:** je matematická funkcie ktorá v neurónovej sieti, ktorá určuje, ako sa vstupy neurónu transformujú na výstup. (najčastejšie: Sigmoid, ReLU, Leaky ReLU, Tanh)
- **Learning rate:** určuje akou veľkou mierou sa aktualizujú váhy modelu počas fázy učenia
- **Epocha:** je proces, počas ktorého neurónová sieť spracuje všetky tréningové dáta práve raz.

## 1.4 Referencie

Github repozitár projektu <https://github.com/TIS2024-FMFI/ml-intro>

Demo zo softvéru Processing (scenáre [1](#), [2](#))

## 1.5 Prehľad nasledujúcich kapitol

Kapitola 2 načrtáva všeobecný popis softvéru. Oboznamuje čitateľa s funkcionalitou Systému a opisuje jeho charakteristiky.

Kapitola 3 definuje špecifické požiadavky týkajúce sa funkčných aspektov a rozhrania.

## 2. Všeobecný popis

### 2.1 Perspektíva produktu

Tento softvér je vyvinutý ako vzdelávací nástroj, ktorý umožňuje používateľom interaktívne pracovať s neurónovými sieťami a získať praktickú skúsenosť so základmi strojového učenia. Softvér je navrhnutý pre akademické účely, najmä pre stredoškolské vzdelávanie, kde sa kladie dôraz na jednoduché a intuitívne pochopenie fungovania neurónových sietí.

Z technického hľadiska je softvér samostatná aplikácia, ktorá využíva simulácie neurónových sietí a ich tréning pomocou algoritmu spätného šírenia. Používatelia môžu meniť parametre sietí a sledovať vplyv týchto zmien na správanie siete, čím získavajú praktickú skúsenosť s procesom učenia.

Softvér je postavený na štandardných základoch strojového učenia a neurónových sietí, ale jeho hlavným cieľom je zjednodušiť tieto koncepty a priblížiť ich širokému publiku.

### 2.2 Funkcie produktu

Po spustení softvéru si používatelia budú môcť vybrať z dvoch preddefinovaných scenárov. Každý scenár bude mať už natrénovanú sieť alebo si používateľ môže natrénovať sieť vlastnú. Scenáre budú [1](#)) rozpoznávanie farieb červená-zelená a [2](#)) trojzložkové rozpoznávanie farieb. Vybraná neurónová sieť sa zobrazí graficky a používateľ bude mať možnosť vidieť štruktúru siete (vrstvy, perceptrony, aktivácie a váhy, ktoré budú reprezentované hranami medzi perceptronami). Táto vizualizácia umožní lepšie pochopenie toho, ako neurónová sieť pracuje a ako sa jednotlivé perceptrony navzájom ovplyvňujú. Používatelia budú môcť zmeniť vstupné parametre (bias, aktivačná funkcia, learning rate) a sledovať, ako tieto zmeny ovplyvňujú výstupné výsledky siete. Softvér umožní používateľom trénovať neurónovú sieť pomocou algoritmu spätného šírenia. Sieť môže byť trénovaná buď na jednom príklade, alebo na celej sade príkladov naraz, pričom používateľ bude môcť sledovať zmeny vo váhach perceptronov a

postupne pozorovať, ako sa sieť zlepšuje pri riešení úloh. Používatelia budú mať možnosť uložiť aktuálny stav neurónovej siete, načítať uložený stav neurónovej siete alebo obnoviť softvér do predvoleného stavu.

## 2.3 Charakteristika používateľov

Softvér nebude rozlišovať medzi rôznymi druhmi používateľov z hľadiska svojej funkcionality, ale bude navrhnutý tak, aby optimalizoval “user experience” pre nasledujúce triedy používateľov.

**Učitelia a inštruktori**, ktorí budú softvér využívať ako nástroj na výučbu. Pre nich je softvér cennou pomôckou pri vysvetľovaní zložitých tém spôsobom, ktorý je zrozumiteľný a prístupný. Učitelia môžu viesť svojich študentov cez jednotlivé scenáre, kontrolovať ich interakcie so softvérom a poskytovať dodatočné vysvetlenia.

**Študenti**, ktorí budú so softvérom pracovať pod vedením učiteľov alebo samostatne v rámci výučby. Softvér je navrhnutý tak, aby bol pre nich intuitívny a poskytoval im okamžitú spätnú väzbu prostredníctvom vizuálnej reprezentácie neurónových sietí. Študenti môžu experimentovať so vstupmi, pozorovať proces učenia a lepšie pochopiť teoretické koncepty v praxi.

## 2.4 Všeobecné obmedzenia

Vývoj tohto softvéru je ovplyvnený niekoľkými dôležitými technickými faktormi, ktoré určujú, ako bude systém navrhnutý a implementovaný. V prvom rade musí byť systém prístupný, jednoduchý na používanie a vhodný pre stredoškolské vzdelávanie. To znamená, že používateľské rozhranie musí byť intuitívne a dostatočne vizuálne, aby bolo prístupné aj študentom bez predchádzajúcich skúseností s neurónovými sieťami alebo programovaním.

Ďalším dôležitým faktorom sú technické obmedzenia. Softvér musí byť schopný bežať na bežných počítačoch, ktoré sú k dispozícii v školských laboratóriách, čo znamená, že aplikácia musí byť optimalizovaná na plynulý chod aj na menej výkonných zariadeniach.

## 2.5 Predpoklady a závislosti

Pri vývoji tohto softvéru predpokladáme, že používatelia budú mať prístup k základnému hardvéru, ako sú bežné školské počítače alebo notebooky, ktoré spĺňajú minimálne Systémové požiadavky. Na strane používateľov predpokladáme základné znalosti práce s počítačom, avšak nie sú vyžadované žiadne špecifické znalosti o programovaní alebo strojovom učení. Celková interakcia so softvérom by mala byť intuitívna a vizuálne jasná.

## 3. Špecifikácia požiadaviek

### 3.1 Základné definície

1. Systém bude realizovaný ako počítačová desktopová aplikácia pre OS Windows verzií 10 a novších.
2. Systém bude mať používateľské rozhranie primárne v anglickom jazyku.

### 3.2 Komponenty

1. Používateľ bude môcť interagovať s toolbarom a ovládacím panelom a renderovacím oknom zobrazujúcim neurónovú sieť

### 3.3 Funkcionalita toolbaru

1. Používateľ bude môcť vybrať jeden z dvoch predvolených scenárov: rozpoznávanie farieb-červená-zelená a trojzložkové rozpoznávanie farieb.
2. Používateľ bude môcť otvoriť na novom tabe v defaultnom prehliadači html súbor s učebným materiálom určeným k zvolenej sieti, ak je nejaká sieť zvolená.
3. Používateľ môže stlačením tlačidla v toolbare zobrazíť hodnoty váh a aktivácií v Rendereri v textovej forme.

### 3.4 Funkcionalita ovládacieho panela

1. Používateľ bude môcť interagovať so vstupom (vkladať vlastné hodnoty), v 1. scenári kliknutím na color picker a v 2. tiež
2. Používateľ bude môcť vybrať [aktivačnú funkciu](#) z predvolených aktivačných funkcií Sigmoid, ReLU, Tanh používanú vo všetkých uzloch siete.
3. Používateľ bude môcť uložiť aktuálny stav (natrenované váhy, parametre siete: počty vrcholov, bias, learning rate, aktuálne nastavená aktivačná funkcia) neurónovej siete
4. Používateľ bude môcť načítať uložený stav neurónovej siete
5. Používateľ bude môcť nastaviť hodnotu biasu
6. Používateľ bude môcť nastaviť [learning rate](#)
7. Používateľ bude môcť spustiť učebný proces, na tréningových dátach, ktoré sú na začiatku náhodne vygenerované ale používateľ ich môže zmeniť.
8. Používateľ bude môcť nastaviť počet [epoch](#) učebného procesu

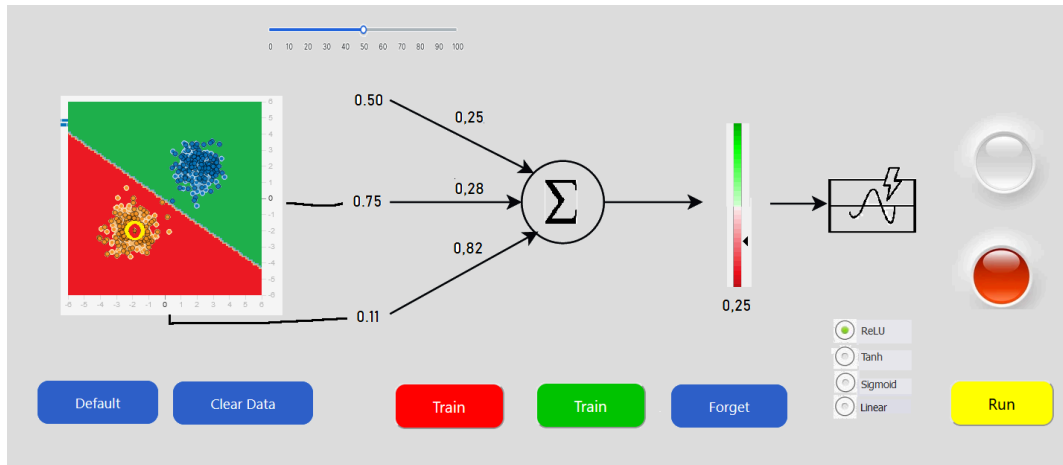
### 3.5 Funkcionalita Renderera

1. Renderovacie okno zobrazí 3d reprezentáciu grafu neurónovej siete
2. Hrany grafu budú vyfarbené podľa ich aktuálnej váhy a priehľadné podľa ich váhy (čím bližšie k 0, tým viac priehľadné)

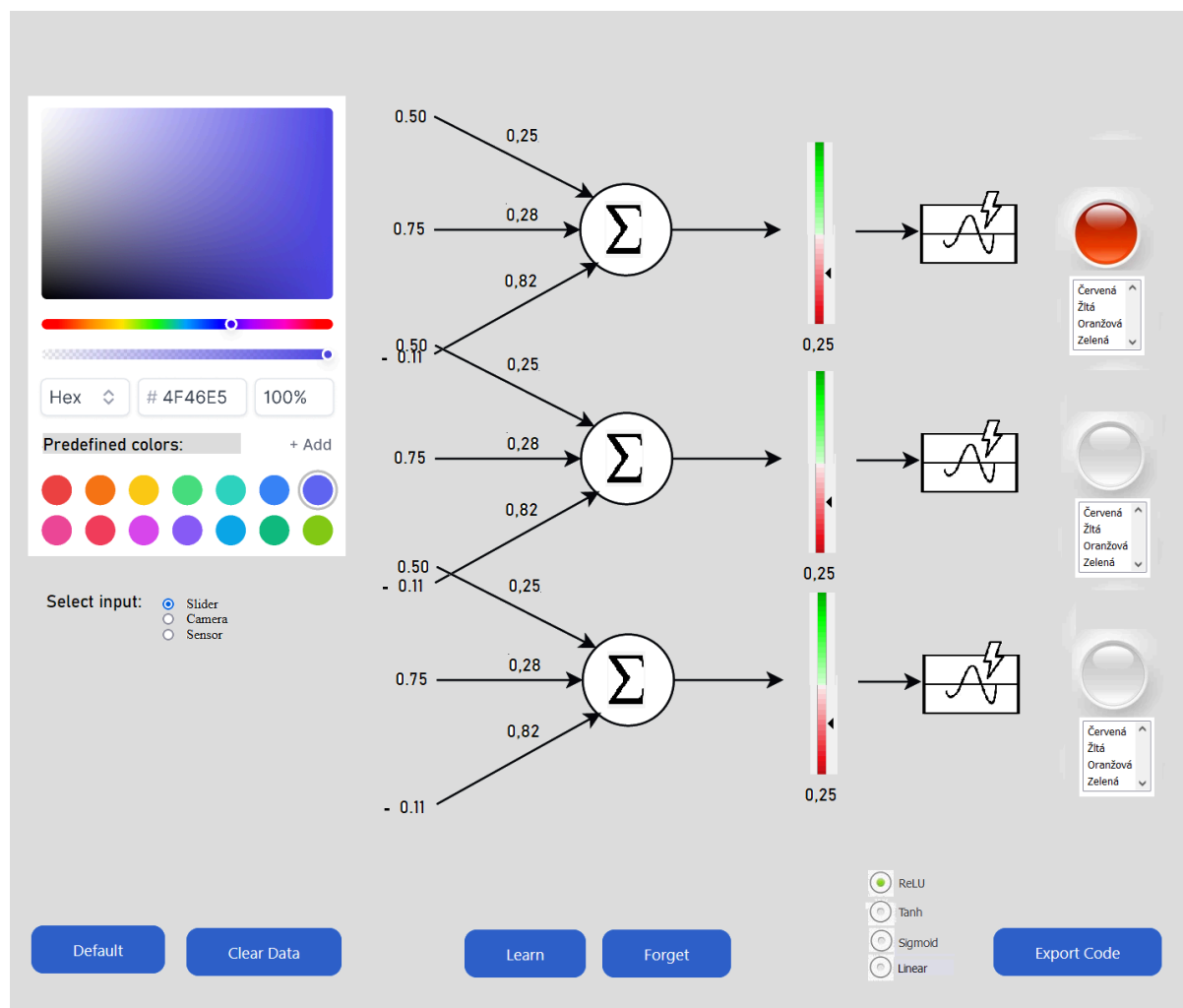
3. Vrcholy grafu budú vyfarbené podľa ich aktuálnej aktivácie
4. Používateľ bude môcť sa pozerať na tento 3d graf
5. Používateľ bude môcť otáčať a približovať tento 3d graf

## 4. Prílohy

### 4.1 Červeno-zelená mapa



## 4.2 Trojzložkové rozpoznávanie farieb

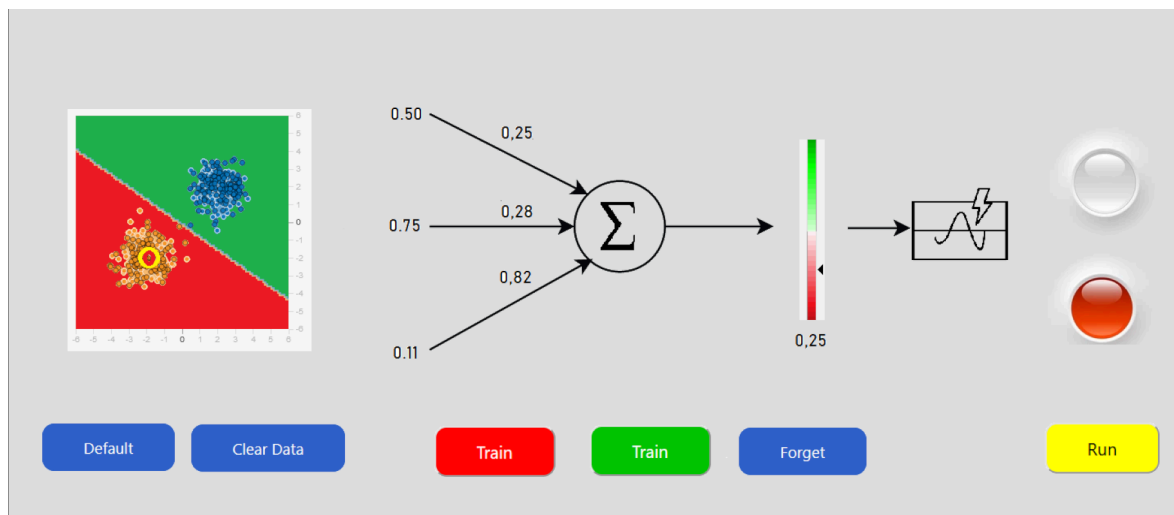




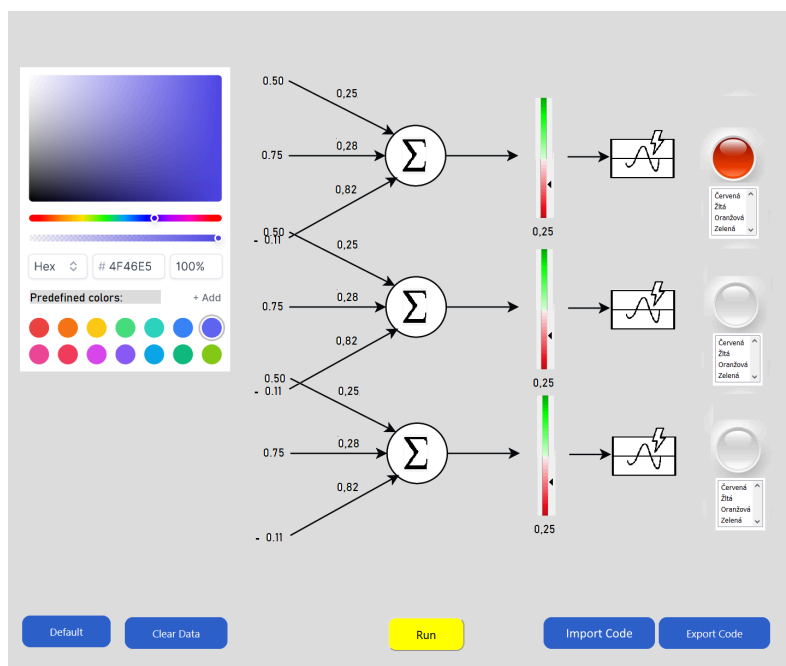
# Návrh

## Návrh používateľského rozhrania

### Scenár Červená-Zelená

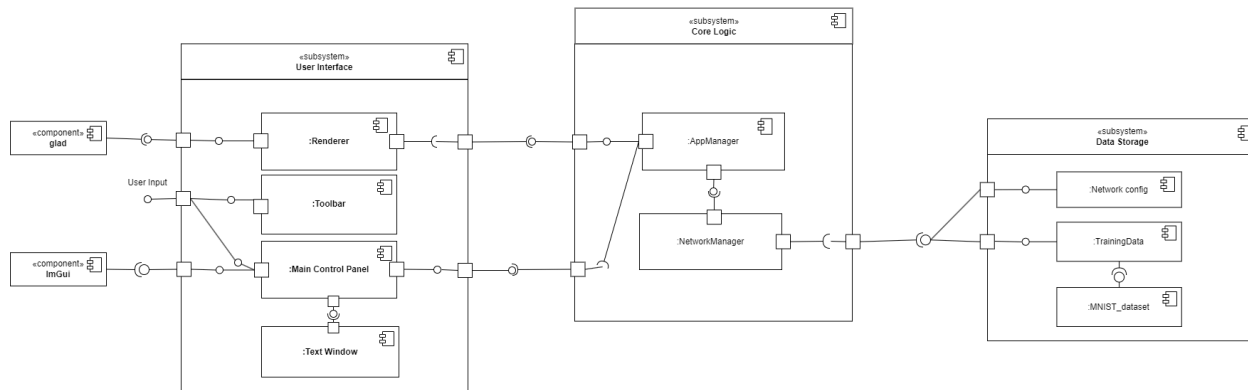


### Scenár RGB

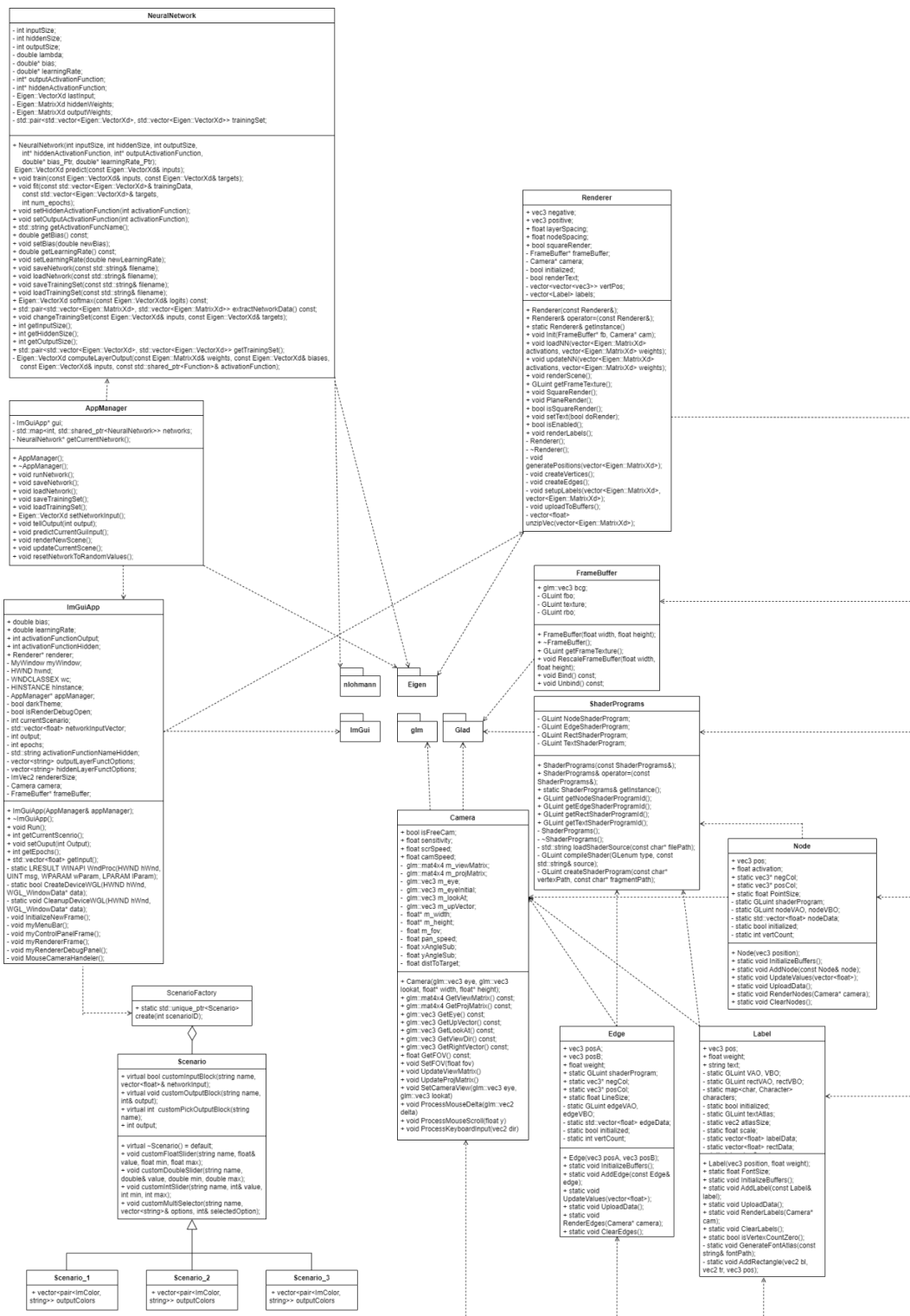


# UML Diagramy

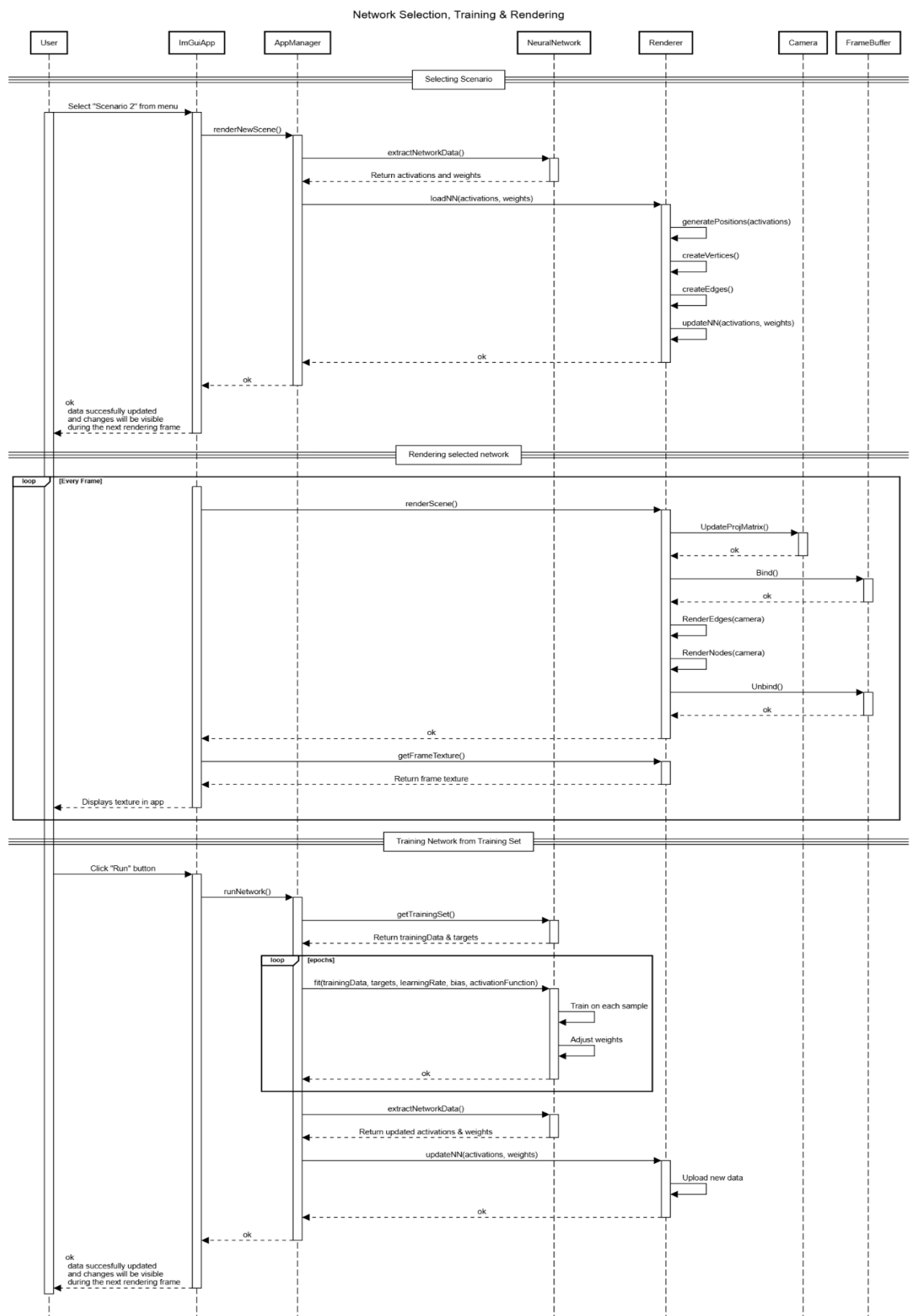
## Component diagram



# Class Diagram



# Sequence diagram



# Podrobný dátový model perzistentných údajov

Uložená neuronová sieť, formát .json:

```
{
  "inputSize": 1,
  "learningRate": 0.05,
  "bias": 1.0,
  "hiddenActivationFunction": "ReLu",
  "hiddenSize": 2,
  "hiddenWeights": [
    [
      0.2776097098355104,
      -0.9309164295100765,
      -0.670940646064215
    ],
    [
      -0.9309164295100765,
      -0.670940646064215,
      2.082005081179753
    ]
  ],
  "outputActivationFunction": "SoftMax",
  "outputSize": 1,
  "outputWeights": [
    [
      -0.5530778758473938,
      -0.3142346106881305
    ]
  ]
}
```

Uložené training data, formát .json:

```
{
  "trainingSet": {
    "inputs": [
      [
        1.0,
        0.0
      ],
      [
        0.0,
        1.0
      ],
      [
        0.3286470424079615,
        0.17430517467050713
      ],
      [
        0.886195162224813,
        0.4175863857200823
      ],
      [
        0.4178367123647131,
        0.9257030122613521
      ]
    ],
    "targets": [
      [
        0.0
      ],
      [
```

```
        1.0
    ],
    [
        0.0
    ],
    [
        0.0
    ],
    [
        1.0
    ]
]
}
}
```

# Plán implementácie

## Fáza 1: Analýza požiadaviek a návrh

1. **Dokončenie požiadaviek**
  - Zhromaždiť podrobné požiadavky, vrátane špecifickej funkcionality pre každý scenár.
  - Dokončiť zoznam závislostí softvéru (knihnice, frameworky a podpora hardvéru).
2. **Návrh architektúry softvéru**
  - Naplánovať architektúru (moduly a UI komponenty).
  - Definovať kľúčové triedy a ich interakcie (UML class diagram):
  - Navrhnuť dátovú štruktúru pre tréning siete a formát na export modelových parametrov. (dátový model perzistentných údajov)
3. **Návrh UI/UX (Hozlar)**
  - Vytvoriť návrhy obrazoviek pre hlavné rozhrania. (GUI)
  - Doplniť 3D grafiku pre zobrazenie neurónových sietí v návrhu UI.

## Fáza 2: Implementácia

### Používateľské rozhranie

1. **Hlavné rozhranie (Hozlár)**
  - Vytvoriť hlavné okno s tlačidlami na otvorenie scénarov
  - Vytvoriť tlačidlo pre otvorenie scenáru
2. **Scenáre (Hozlár)**
  - Vytvoriť okno pre každý scenár
  - V každom scenári vytvoriť tlačidlo "späť"
3. **Vysvetľujúce okná (Miškovčik)**
  - Vytvoriť modul ktorý bude spravovať vysvetľujúce okná
  - Pridať tlačidlo pre každý scenár ktoré bude obsahovať popis scenáru
4. **Ovládacie Prvky pre Úpravu Parametrov a Tréning (Kováč)**
  - Vytvoridsetrik tréningu a začatie/ukončenie tréningu.
  - Zabezpečiť, aby používatelia mohli prepínať medzi manuálnym krokovým tréningom a režimom plynulého tréningu.

### Backend

1. **Modul pre štruktúru neurónových sietí (Miškovčik)**
  - Vytvoriť classu pre perceptron
    - Inputy



- Outputy
- Váhy
- Vytvoriť modul pre neurónové siete:
  - Modul bude využívať spravenú classu perceptron
  - Inicializáciu váh
  - Dopredné šírenie
  - Algoritmus backpropagation, s možnosťou krokovať
  - Aktivačné funkcie a úpravu parametrov (rýchlosť učenia, počet vrstiev)
  - Vytvoriť funkcionality exportu siete
- 2. **Renderer pre neurónové siete** (Klieštik)
  - Vytvoriť vizuálny komponent na zobrazenie neurónov
  - Vytvoriť vizuálny komponent na zobrazenie z neurónov vrstiev
  - Vytvoriť vizuálny komponent na zobrazenie váh
  - Vytvoriť vizuálny komponent na zobrazenie aktivácií.
  - Aktualizovať zobrazenie siete po každom kroku
- 3. **Scenáre** (Hozlár, Miškovčík)
  - Natrénovať jednu sieť pre každý scenár
  - Vytvoriť tlačidlo do každého scenáru ktorou su používateľ načíta už natrénovanú sieť
  - Vytvoriť tlačidlo na nastavenie parametrov pre každý scenár
  - Vytvoriť tlačidlo na nastavenie parametrov začatie/ukončenie tréningu pre každý scenár
  - Vytvoriť tlačidlo export vytvorenej siete pre každý scenár
  - **Klasifikácia Červeno-Zelenej mapy**
    - Vytvoriť modul na ktorom si používateľ vyberie bod z červeno-modrého gradientu a modul ho prekonvertuje na vstup pre neurónovú sieť
    - Vytvoriť tlačidlá ktorými používateľ bude hovoriť výsledky pre sieť, teda ju učiť (tlačidlá "červená" a "modrá")
  - **Rozpoznávanie RGB komponentov**
    - Vytvoriť modul so vstupom ako RGB komponenty ktoré si zvolí používateľ a prekonvertuje na vstup pre neurónovú sieť

## Fáza 4: Testovanie

1. **Jednotkové testovanie** (Kováč)
  - Napísať testy pre perceptron
  - Napísať testy pre model neurónovej siete
  - Testovať komponenty užívateľského rozhrania
2. **Integračné testovanie** (Klieštik, Miškovčík, Hozlár)
  - Testovať integráciu všetkých modulov medzi sebou
  - Overiť, že každý scenár funguje podľa očakávaní

### 3. **Testovanie použiteľnosti** (Kováč)

- Uskutočniť testovanie použiteľnosti, odsimulovať skutočné použitie

## **Fáza 5: Finalizácia** (Klieštik, Miškovčík, Hozlár, Kováč)

### 1. **Konečné úpravy**

- Odkonzultovať výsledný software so zadávateľom a upraviť podľa spätnej väzby
- Opraviť všetky chyby nájdené pri testovaní

### 2. **Dokumentácia**

- Dopísať akékoľvek zmeny v dokumentácii