# KIV/PPR

Adam Mištera

# Architektura aplikace

- Aplikace napsaná v C++*17*
- 3 hlavní třídy aplikace
- Třída *Matrix*
  - Reprezentace matice
  - Všechny potřebné operace
- Třída *NeuralNetwork*
  - Kód neuronové sítě
  - V této třídě probíhá trénink
- Třída *Analyzer*
  - Analýza výsledků a generování grafu
- Pomocné třídy a struktury
  - *Loader*, *Record*, *Saver*, *Settings*

# Třída *Matrix*

- Třída reprezentující matici
- Data uložená pomocí *std::vector*
- Základní operace s maticemi
  - Sčítání, odčítání, násobení, transponování
- Přetížené operátory pro základní operace
  - (), +, -, *, /
  - Přístup k prvku $a_{ij}$ matice A pomocí A(i, j)
- Mnoho užitečných metod
  - Funkce *tanh*, *relu*, *argmax*, *max* a další
  - Generování náhodné matice

# Třída *NeuralNetwork*

- Obsahuje implementaci neuronové sítě
  - Včetně backpropagation
  - Stochastic Gradient Descent (SGD)
- Vrstvy 8 – 16 – 26 – 32
  - *Tanh, tanh* a *softmax*
  - Multiclass klasifikace
  - Využívá risk funkci pro normalizaci vstupu
- Ve finální verzi experimenty s použitím SGD s momentem
  - V případě příznivé časové situace také s Adaptive Moment Estimation (ADAM)

# Třída *Analyzer*

- Generování grafu neuronové sítě
- Spočtení statistiky relativní chyby
- Uložení statistik do CSV souboru
- Implementace zatím není kompletní
  - Chybí výstup do grafu

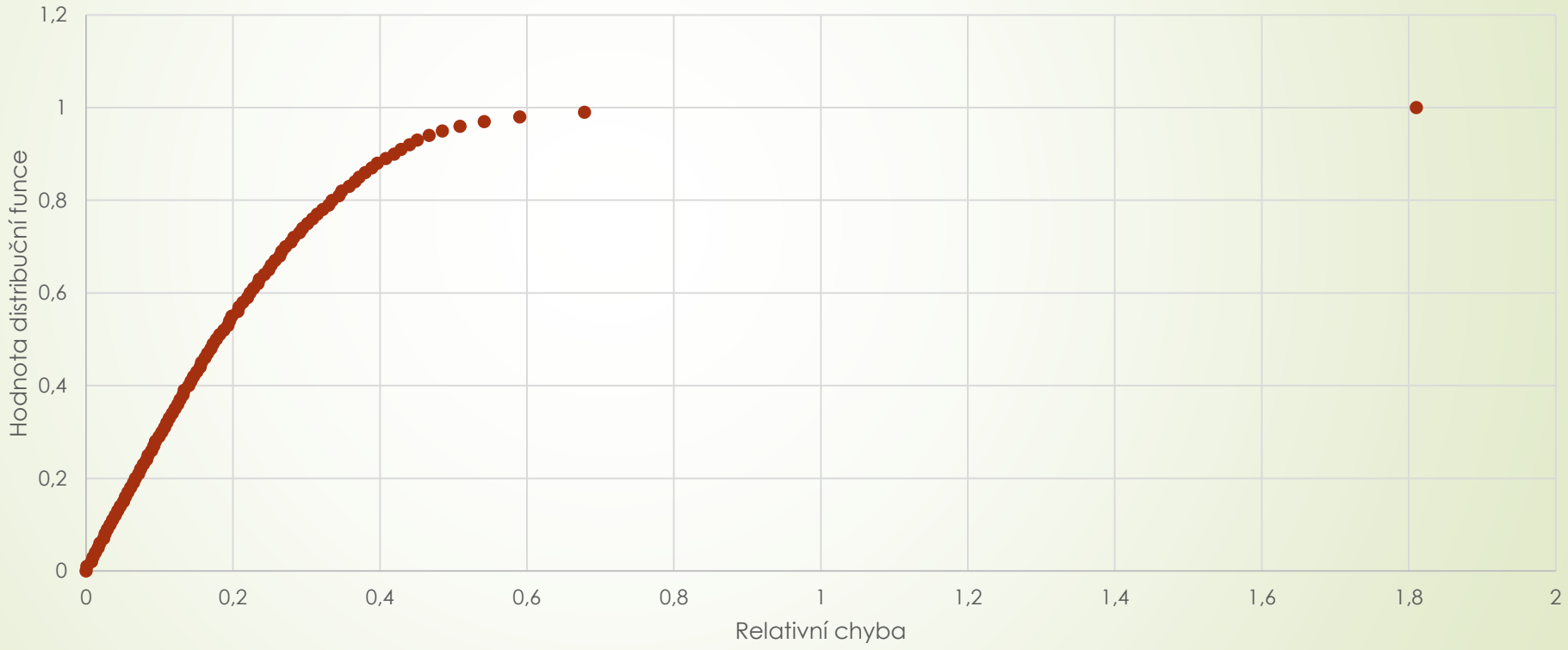
# Testování modelu

- Datová sada rozdělena
  - 60 % trénink, 40 % testování
- Predikce pro dané *t*
  - 30, 60, 90, 120 a 240 minut
- 10 pokusů – výsledky zprůměrované
- Velmi dobré výsledky pro předpověď na 30, 60 a 90 minut

# Výsledky tréninku

| *t* | Chyba - Trénovací sada | Chyba - Testovací sada |
|---|---|---|
| 30 min | **22.6 %** | **23.5 %** |
| 60 min | 30.5 % | 30.9 % |
| 90 min | 33.1 % | 33.6 % |
| 120 min | 33.5 % | 42.0 % |
| 240 min | 35.3 % | 51.9 % |

# Empirická kumulativní distribuční funkce relativní chyby pro t = 30 minut



# Paralelní kód

- Využití C++ PSTL
- Trénování více instancí modelů najednou
  - Průměrování nejlepších parametrů
- Testováno pod Windows 10 Pro
  - i7 – 6500U – Skylake, 2.5 GHz, 2 jádra, 4 vlákna
  - Překladač MSVC 19.25
  - 64bit, release mode, zapnuté optimalizace
- Do finální verze
  - Testování pod procesorem AMD (Zen 2 - 4/8)

# Naměřené výsledky

| | 10 modelů | 25 modelů | 100 modelů |
|---|---|---|---|
| Sériově | 9401 ms | 24 015 ms | 69 789 ms |
| Paralelně | 4002 ms | 10 692 ms | 31 815 ms |
| Urychlení | ≈ 2.35 | ≈ 2.25 | ≈ 2.19 |

# Další postup

- Dokončení kódu pro vykreslení grafu neuronové sítě
- Dokončení kódu pro asymetrický multiprocessor
  - OpenCL vs. C++ AMP
  - C++ AMP – jednodušší, ale dále nevyvíjené
  - OpenCL – komplikovanější, ale hojně používané

Děkuji za pozornost

Nyní je prostor pro Vaše dotazy