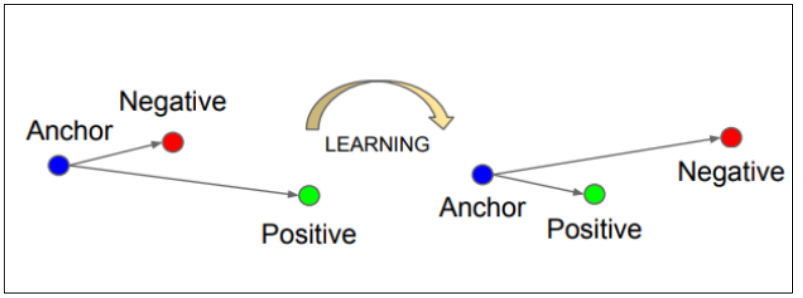
Projeto PSI3501 – Verificação automática da voz - P2

Piero Conti Kauffmann (8940810)

Introdução

Triplet Loss

- Composição da tripla *(A, P, N)*
  - Instância chave (*A*) de um usuário *i*
  - Instância positiva (P) do mesmo usuário *i*
  - Instância negativa (N) de um outro usuário *j*



Retirado de Schroff et. al (2015)

- **Objetivo**: Aprender a função *g: R^N → R^D (D << N)* que mantém *g(A)* mais perto de *g(P)* do que *g(N)*, com uma margem *m* de segurança

$$\|g(x_i^a) - g(x_i^p)\|_2^2 + m < \|g(x_i^a) - g(x_j^n)\|_2^2, \ i \neq j$$

Função de custo

Função de custo (tri)

$$\mathcal{L}_{tri}(\theta) = \sum_{i \neq j} \sum_{\substack{a,p,n \\ a \neq p}} [m + \|g_\theta(x_i^a) - g_\theta(x_i^p)\|^2 - \|g_\theta(x_i^a) - g_\theta(x_j^n)\|^2]_+$$

Todas as triplas possíveis

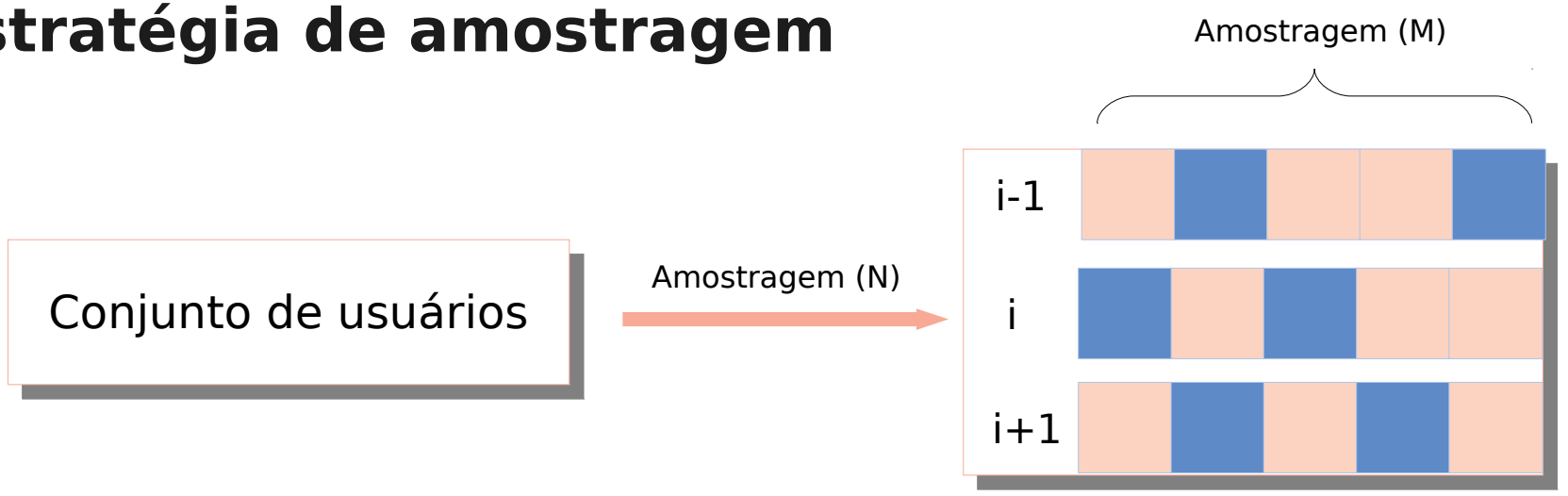
Onde $[.]_+$ é a função *ReLU(x) = max(0, x)*

- **Dificuldade principal**: como escolher as triplas?
  - Rápida obtenção
  - Informações relevantes para o aprendizado

Função de custo

- **Dificuldade principal**: como escolher as triplas?
  - Rápida obtenção
  - Informações relevantes para o aprendizado

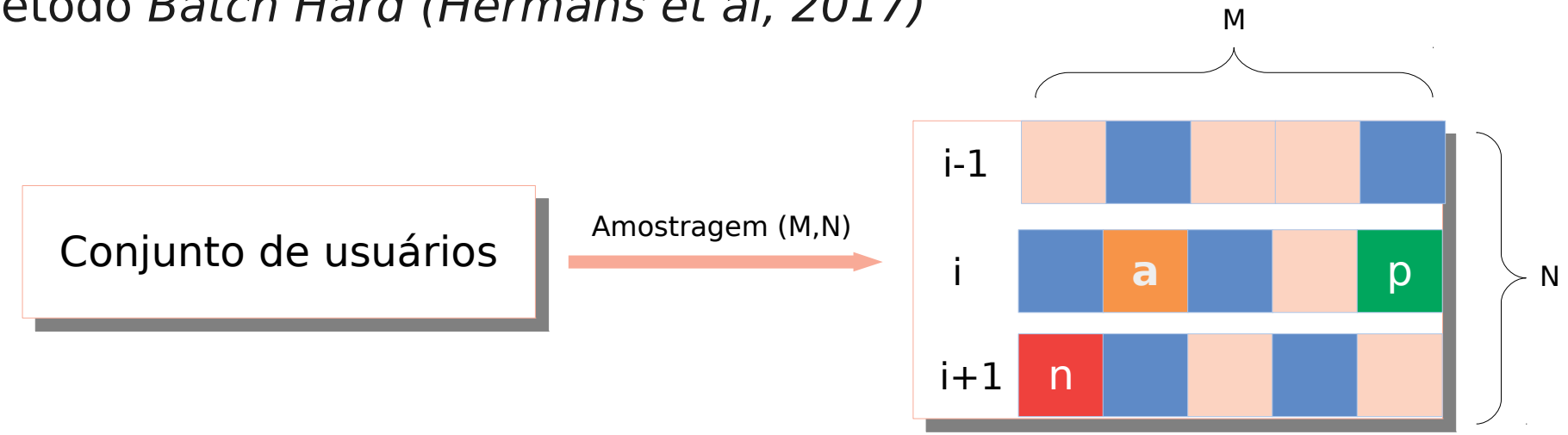
Estratégia de amostragem



Função de custo

- **Dificuldade principal**: como escolher as triplas?

  Método *Batch Hard (Hermans et al, 2017)*

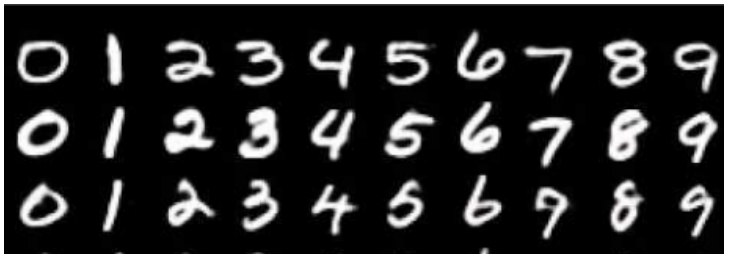


Positivo mais distante — Negativo mais próximo

$$\mathcal{L}_{BH}(\theta) = \sum_{i=1}^{M}\sum_{a=1}^{N}[\, m + \max_{p=1,2,\dots N} \|g_\theta(x_i^a) - g_\theta(x_i^p)\|^2 - \min_{\substack{j=1,\dots,M \\ n=1,\dots,N \\ j\neq i}} \|g_\theta(x_i^a) - g_\theta(x_j^n)\|^2 \,]_+$$

Datasets – Casos de uso

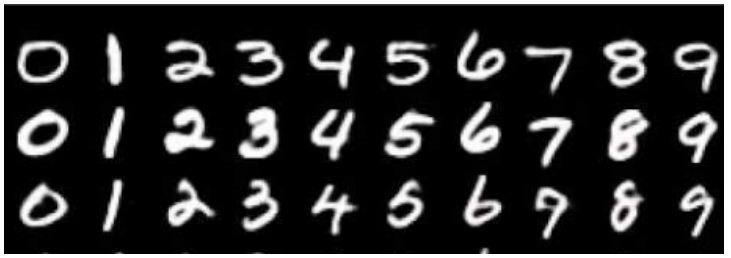
- MNIST
  - 70 mil Imagens de dígitos (0-9) manuscritos
  - Avaliar cada “dígito” como um “usuário”
  - Útil para prototipagem



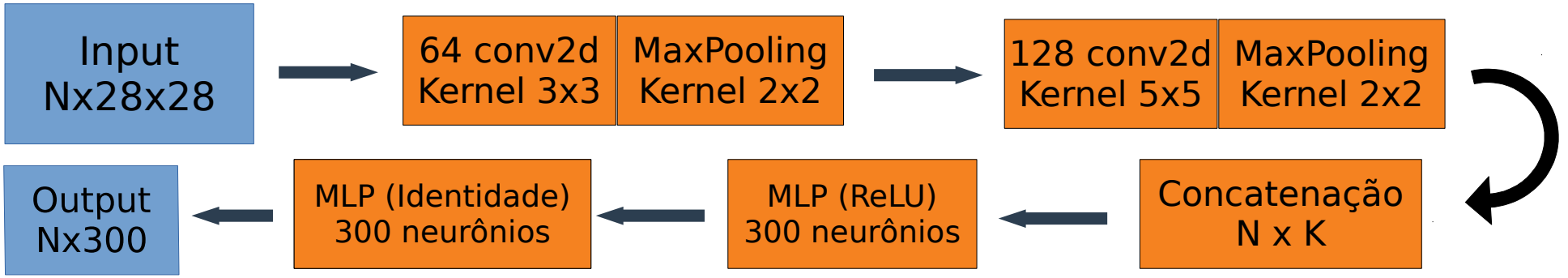
- Gravações de usuários do site ‘VoxForge’
  - Gravações baixadas por *script* do site voxforge.org (licenciados sob a GNU/ GPL de uso livre)
  - 195 usuários, cada um com uma média de 15 trechos
  - Cada trecho tem uma duração média de 12 segundos
  - *Text Independent*

Base de dados MNIST

- MNIST
  - Treino: 60 mil imagens
  - Teste: 10 mil imagens
  - Representação: imagens 28x28

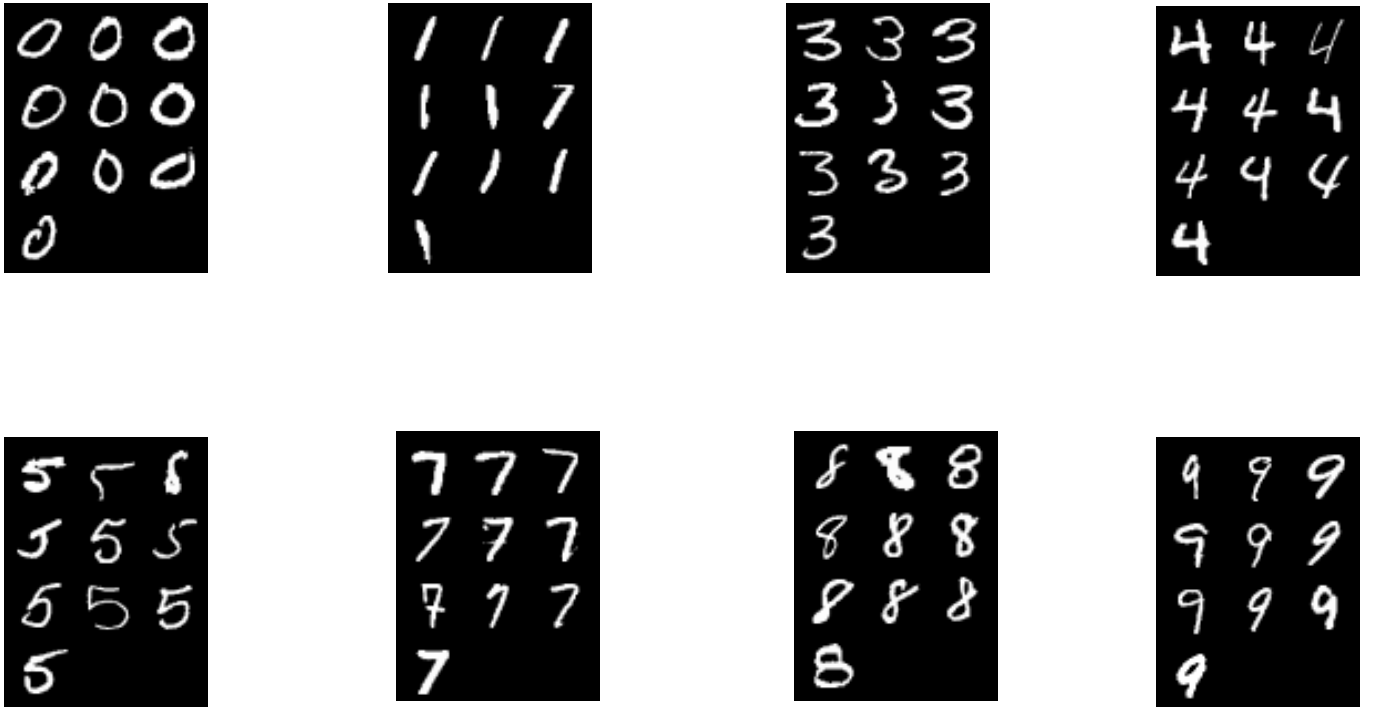


- Arquitetura da rede neural *g (CNN)*



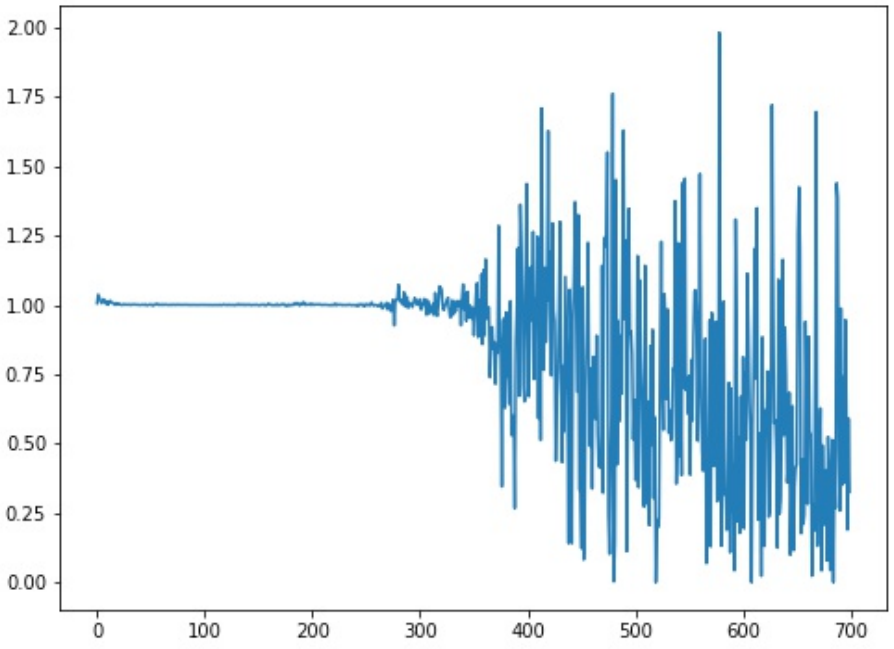
Batch de exemplo

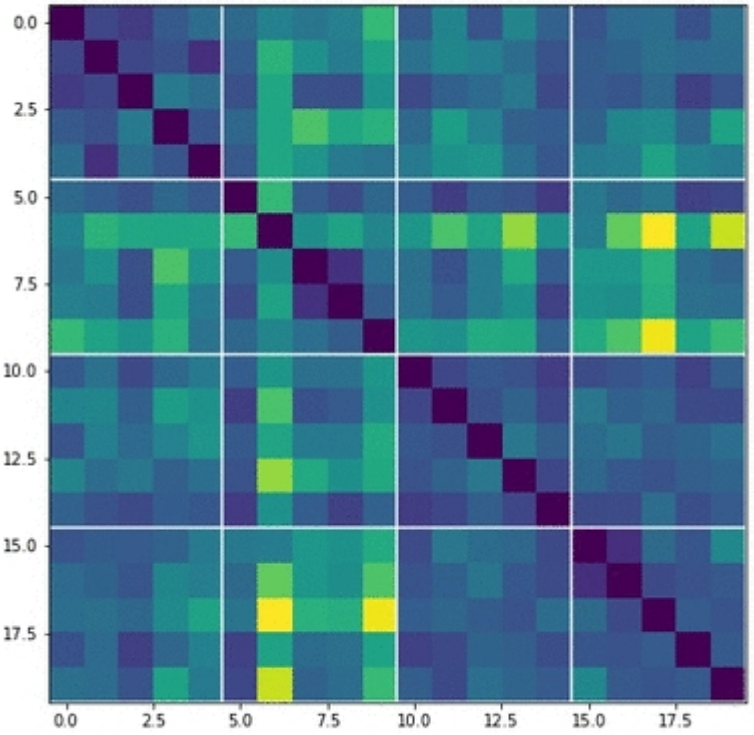
- **Exemplo de um batch com N = 8 (‘usuários’), M = 10 (instâncias)**



Treinamento

- 700 iterações, N = 4 usuários, M = 5 instâncias
- Custo ‘batch-hard’

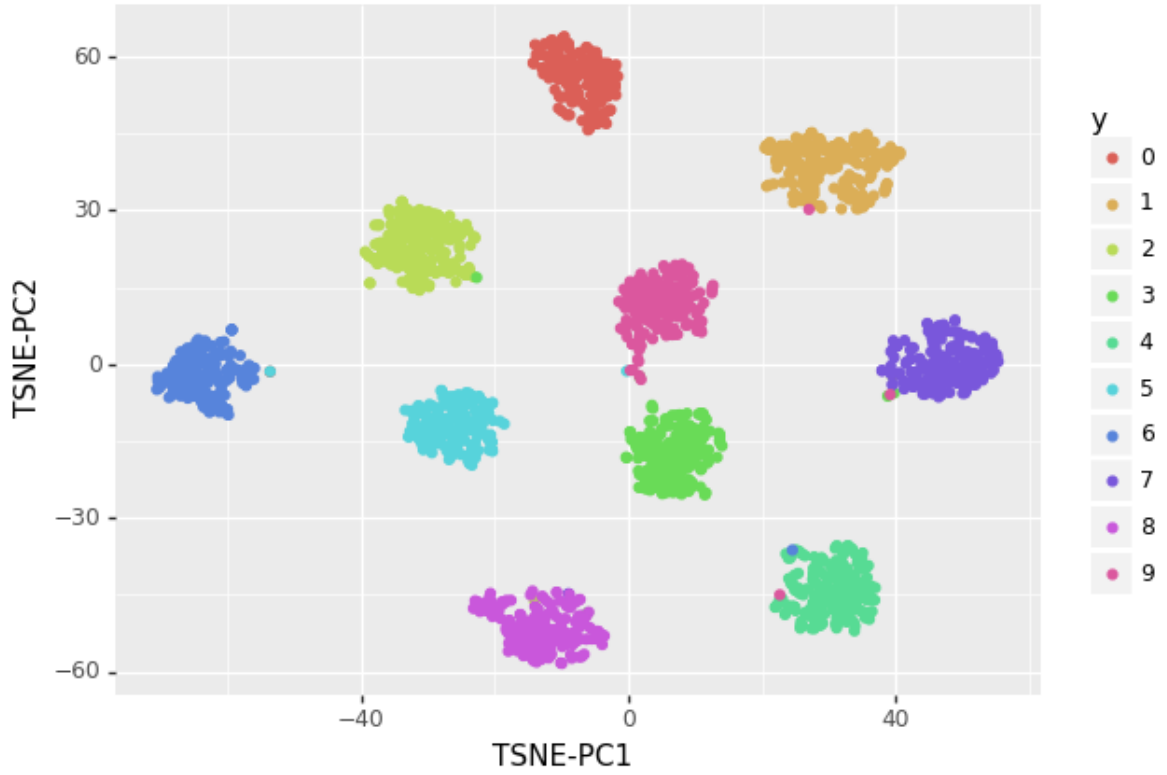




Performance

- Aplicação da rede neural g no conjunto de testes
  - Vetores obtidos com D = 300 dimensões

Aplicamos um método (T-SNE) de redução de dimensionalidade para visualizar os vetores dos dados de teste



Base de dados VoxForge

- VoxForge
  - *wav* de 22050 Hz

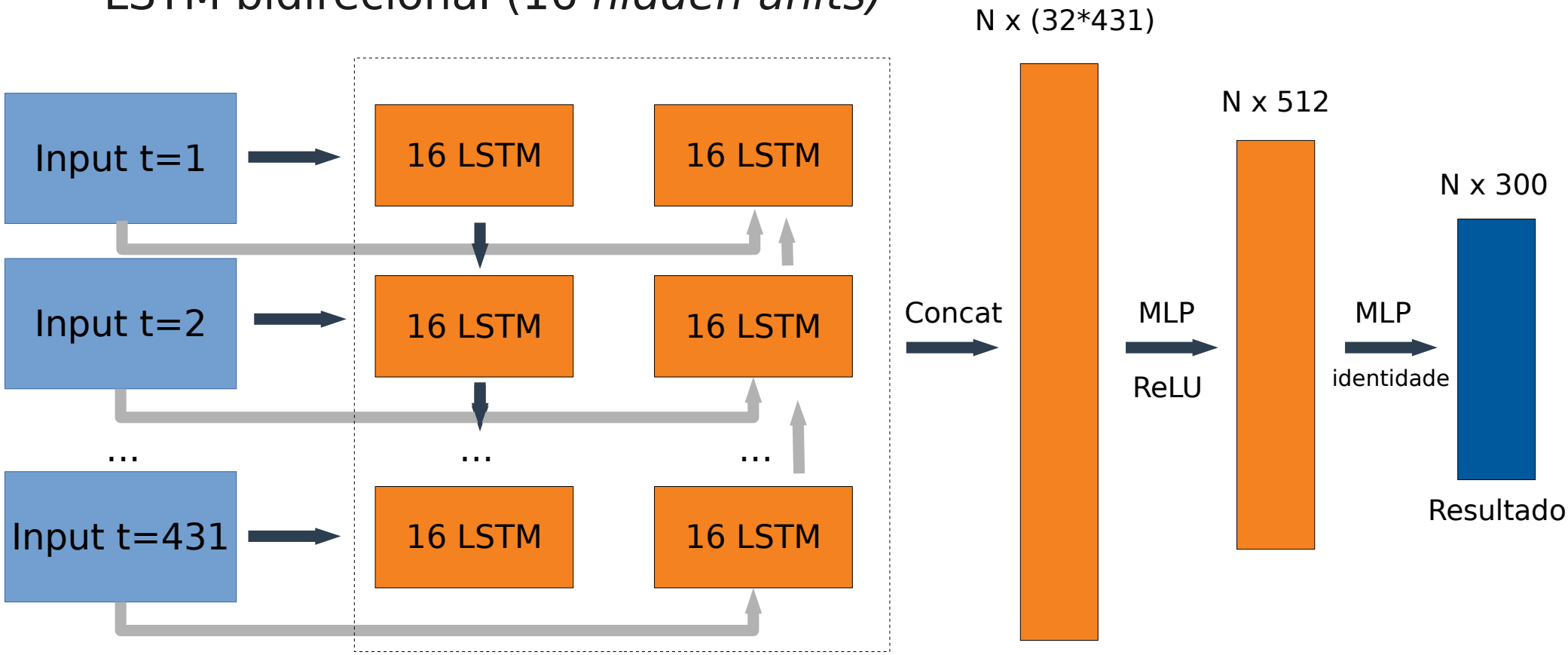
- Pré processamento e extração de *features*
  - Corte ou extensão de cada arquivo, tal que todos possuam 10 segundos
  - Extração de 20 coeficientes Mel-Cepstrais (MFCC)

    Resultado: X: N (*batch-size*) x 20 (*features*) x 431 (tempo)

  - Cálculo do Z-score de cada um dos coeficientes Mel-Ceptrais para que estes possuam média zero e desvio-padrão unitário
    - X[k, i, :] = (X[k, i, :] - média(X[:, i, :])) / desvio-padrão(X[:, i, :])

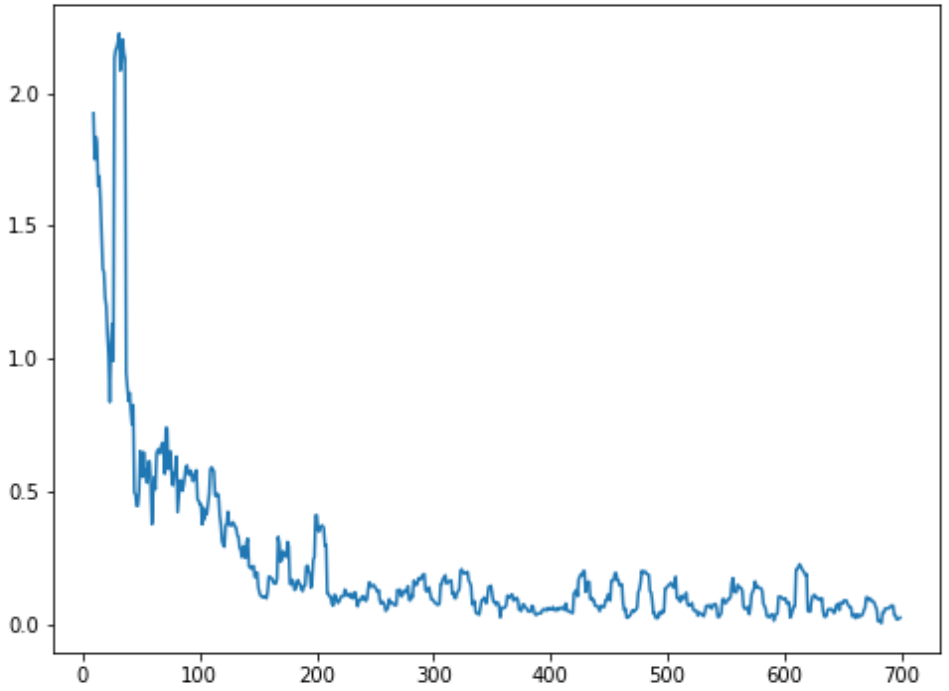
Base de dados VoxForge

- Arquitetura da rede neural
  - LSTM bidirecional (16 *hidden units)*



Treinamento

- Figura: média móvel (janela de 10 pontos) da função de custo (método *batch-hard com N = 10 e M = 5*)



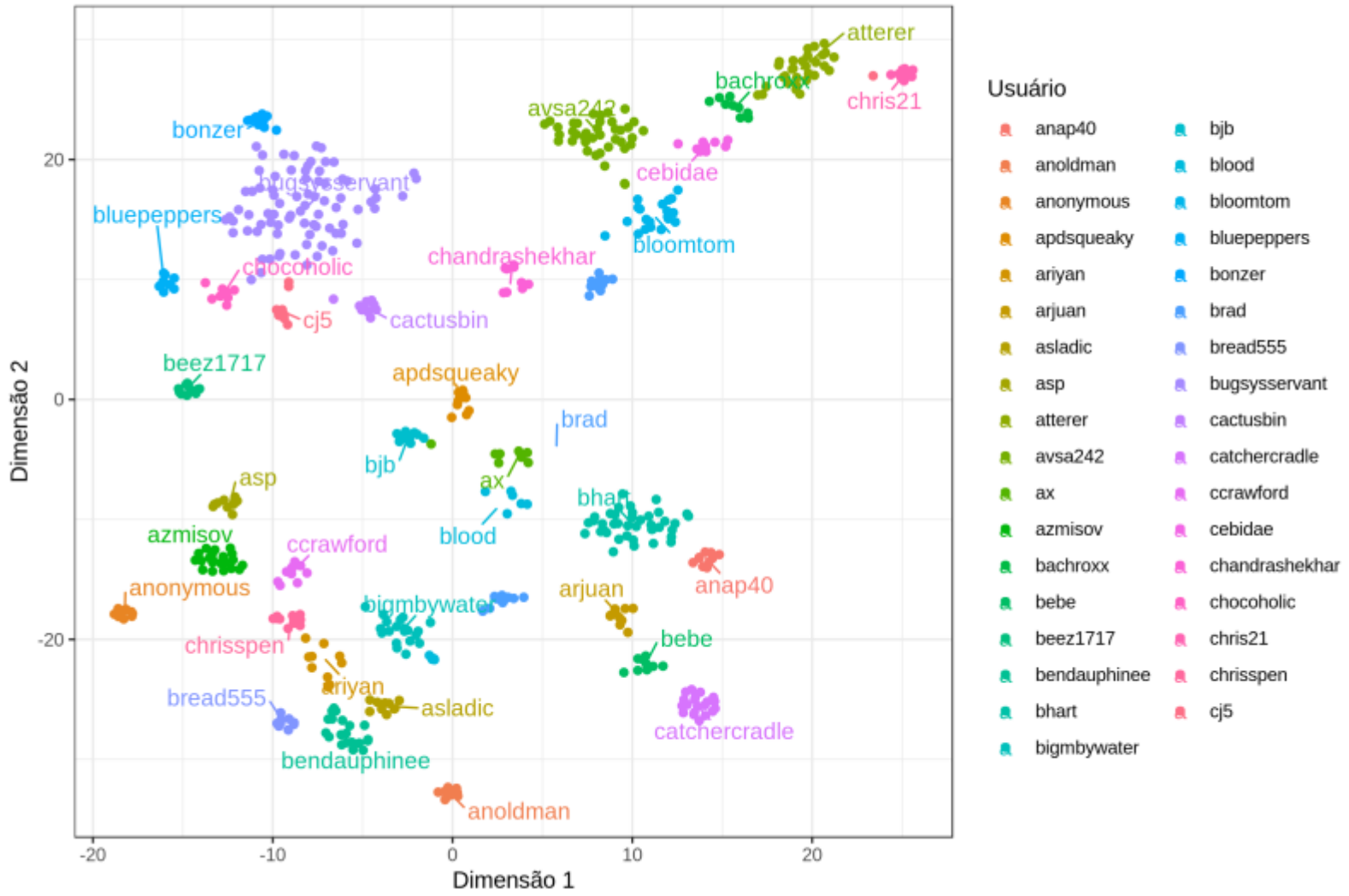
Conjunto de teste

- 36 usuários não vistos previamente pelo modelo

Lista de usuários do conjunto de teste com o respectivo número de amostras:

anap40 (10)
anoldman (10)
anonymous (10)
apdsqueaky (10)
ariyan (10)
arjuan (10)
asladic (10)
asp (10)
atterer (30)
avsa242 (39)
ax (10)
azmisov (20)
bachroxx (10)
bebe (10)
beez1717 (10)
bendauphinee (20)
bhart (40)
bigmbywater (20)
bjb (10)
blood (10)
bloomtom (20)
bluepeppers (10)
bonzer (10)
brad (20)
bread555 (10)
bugsysservant (80)
cactusbin (10)
catchercradle (20)
ccrawford (10)
cebidae (10)
chandrashekhar (10)
chocoholic (9)
chris21 (10)
chrisspen (10)
cj5 (10)

Redução de dimensionalidade (TSNE) dos vetores preditos no conjunto de teste



Performance dos vizinhos mais próximos do conjunto de teste

- Método
  - Para cada instância j do conjunto de teste, avaliam-se quais usuários que produziram os K vizinhos mais próximos de j
  - Caso o usuário que produziu j seja o mesmo da maioria dos seus K vizinhos, é registrado um ‘acerto’, caso contrário, um ‘erro’

- Taxa de acertos em função de K

	K = 1	K = 3	K = 5
Taxa de acertos (%)	100%	99,64%	99,74%