

Universidade Federal de São Carlos

**Bruna Zamith Santos
Elaine Cecília Gatto**

Comparação das Redes Neurais Restricted Boltzmann Machine,
Autoencoder e Denoising Autoencoder para recuperação dos dígitos
MNIST

Trabalho apresentado à disciplina de In-
trodução a Redes Neurais Artificiais do
curso de Ciência da Computação do De-
partamento de Computação da Univer-
sidade Federal de São Carlos.

Professor Dr. Ricardo Cerri.

**São Carlos
Julho
2018**

Conteúdo

| | | |
|----------|--|----------|
| 1 | Introdução e objetivos | 2 |
| 2 | Configuração das Redes Neurais e Implementações | 2 |
| 2.1 | Restricted Boltzmann Machine | 2 |
| 2.2 | Autoencoder | 3 |
| 2.3 | Denoising Autoencoder | 3 |
| 3 | Metodologia de Comparação | 3 |
| 4 | Resultados | 4 |
| 4.1 | Execução 1 | 4 |
| 4.2 | Execução 2 | 4 |
| 4.3 | Execução 3 | 6 |
| 4.4 | Execução 4 | 7 |
| 4.5 | Execução 5 | 7 |
| 4.6 | Execução 6 | 7 |
| 4.7 | Execução 7 | 8 |
| 4.8 | Execução 8 | 9 |
| 4.9 | Execução 9 | 11 |

1 Introdução e objetivos

Redes Neurais Artificiais são modelos matemáticos quem simulam os neurônios biológicos e suas sinapses. Tais redes são capazes de aprender padrões, classificar e agrupar dados através de treinamento e ajustes matemáticos[1].

Neste relatório serão abordados três tipos de implementações de Redes Neurais: (i) Restricted Boltzmann Machine (RBM); (ii) Autoencoder; e (iii) Denoising Autoencoder.

Como tais redes neurais podem ser utilizadas para recuperação de padrões, após um treinamento prévio, este trabalho tem como objetivo ajustar cada tipo de implementação para recuperar dados corrompidos do conjunto de dados MNIST¹, fazendo comparações entre os resultados obtidos. MNIST é um banco de dados de números escritos a mão e em escalas de cinza, como expoto na Figura 1.



Figura 1: Exemplo de dados do conjunto MNIST

O presente trabalho está organizado e apresentado da seguinte forma: na Seção 2 vê-se a configuração e detalhes de implementação de cada rede neural. A Seção 3 aborda a metodologia de comparação utilizada no desenvolvimento deste trabalho. Por fim, na Seção 4 é apresentado os resultados obtidos.

2 Configuração das Redes Neurais e Implementações

Para a execução das comparações entre as redes neurais para com o conjunto de dados MNIST, foram definidas configurações e estratégias para cada arquitetura de rede, as quais são descritas nas subseções seguintes.

2.1 Restricted Boltzmann Machine

Para a configuração da RBM foram utilizados 784 neurônios na camada de entrada, correspondendo ao tamanho do vetor de entrada; e 783 neurônios na camada escondida (entrada - 1). Ainda, usamos a função de ativação sigmoideal; taxa de aprendizagem igual 0.2; K (*Gibb's Sampling*) igual a 1; e 100 épocas. A configuração é ilustrada na Tabela 1.

Tabela 1: Configuração da Restricted Boltzmann Machine

| Qtd neurônios por camada | | | | | |
|--------------------------|-----------|---------------|------------|---|--------|
| Entrada | Escondida | Fun. Ativação | T. Aprend. | K | Épocas |
| 784 | 783 | Sigmoideal | 0.2 | 1 | 100 |

¹<http://yann.lecun.com/exdb/mnist/>

2.2 Autoencoder

Na configuração do Autoencoder utilizou-se 784 neurônios na camada de entrada, correspondendo ao tamanho do vetor de entrada; N neurônios na camada escondida ($784+(30/70/90)$) e 784 neurônios na camada de saída. Sendo o a função de ativação sigmoidal e taxa de ruído M (0.2/0.3/0.5). Ainda, definimos como taxa de aprendizado o valor 0.2 e como limiar para o erro, 0.1. A configuração pode ser observada na Tabela 2.

Tabela 2: Configuração do Autoencoder

| Qtd neurônios por camada | | | | | | |
|--------------------------|-----------|-------|---------------|----------|------------|--------|
| Entrada | Escondida | Saída | Fun. Ativação | T. Ruído | T. Aprend. | Limiar |
| 784 | N | 784 | Sigmoidal | M | 0.2 | 0.1 |

Optamos por uma arquitetura de Autoencoder com Camada Oculta Sobrecompleta para podermos avaliar a existência ou não de *overfitting* dos dados após o treinamento.

2.3 Denoising Autoencoder

E por fim para o Denoising Autoencoder, foram utilizados os mesmos parâmetros que o Autoencoder, como pode ser visto na Tabela 3. As diferenças fundamentais entre o Denoising Autoencoder e o Autoencoder são: O Autoencoder é treinado com os padrões originais do MNIST, enquanto que o Denoising Autoencoder é treinado com padrões corrompidos (função “corrompe.padrão”); no Autoencoder, o erro é calculado entre a saída e o padrão inserido, já no Denoising Autoencoder, o erro é calculado entre a saída e o padrão original (não o corrompido, que foi inserido).

Tabela 3: Configuração do Denoising Autoencoder

| Qtd neurônios por camada | | | | | | |
|--------------------------|-----------|-------|---------------|----------|------------|--------|
| Entrada | Escondida | Saída | Fun. Ativação | T. Ruído | T. Aprend. | Limiar |
| 784 | N | 784 | Sigmoidal | M | 0.2 | 0.1 |

Deste modo, as arquiteturas do Autoencoder e do Denoising Autoencoder são iguais, mas a implementação e os padrões dados como entrada são diferentes.

3 Metodologia de Comparação

Considerando o objetivo de comparar os 3 algoritmos na recuperação dos padrões do conjunto de dados MNIST, foram feitas as seguintes variações:

- RBM variando com 10, 50 e 100 padrões de entrada
- Autoencoder e Denoising Autoencoder variando com:
 - 10, 50 e 100 padrões de entrada;
 - 30, 60 e 90 neurônios a mais na camada escondida;
 - 0.2, 0.3 e 0.5 de taxa de ruído.

Para as análises, plotamos os resultados para cada uma das execuções. Dado o grande volume de resultados, optamos por anexá-los ao relatório (junto ao código fonte) e aqui apenas expor os mais relevantes. Eles encontram-se na próxima seção.

4 Resultados

Os parâmetros foram variados um por um. As execuções de 1 a 3 variam o parâmetro "Número de neurônios a mais" na camada escondida, e os outros parâmetros são mantidos fixos. As execuções de 4 a 6 variam o parâmetro "taxa de ruído" e os outros parâmetros são mantidos fixos. Por fim, as execuções de 7 a 9 variam o parâmetro "número de imagens carregadas" e os outros parâmetros são mantidos fixos.

4.1 Execução 1

- Número de neurônios a mais = 30
- Taxa de Ruído = 0.3
- Imagens = 50

Nesta configuração a rede RBM teve pior desempenho e a Autoencoder melhor desempenho. A rede Denoising Autoencoder ficou no meio termo.

4.2 Execução 2

- Número de neurônios a mais = 60
- Taxa de Ruído = 0.3
- Imagens = 50

Nesta configuração nota-se que que todas as redes conseguem desenhar com maior precisão os dígitos, porém, o que muda claramente é a quantidade de pixels brancos printado. Portanto, com 60 neurônios a mais, a reconstrução da entrada original melhora para todos. Ainda assim, entre as redes, a AutoEncoder continua com o melhor resultado, a RBM com pior e Denoising Autoencoder mediano.

As Figuras 2, 3 e 4 expõem os resultados para o RBM, Autoencoder e Denoising Autoencoder, respectivamente, da execução 2.

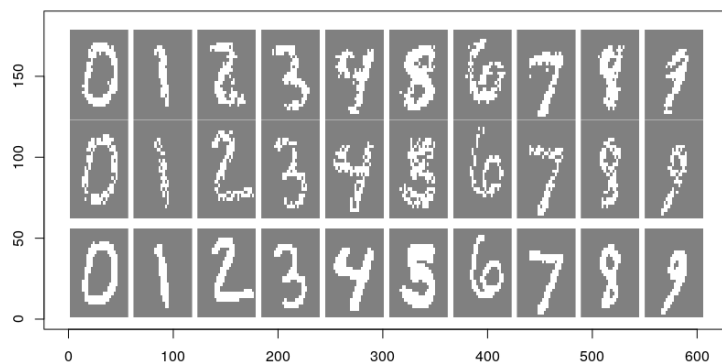


Figura 2: Resultados obtidos para a RBM na execução 2

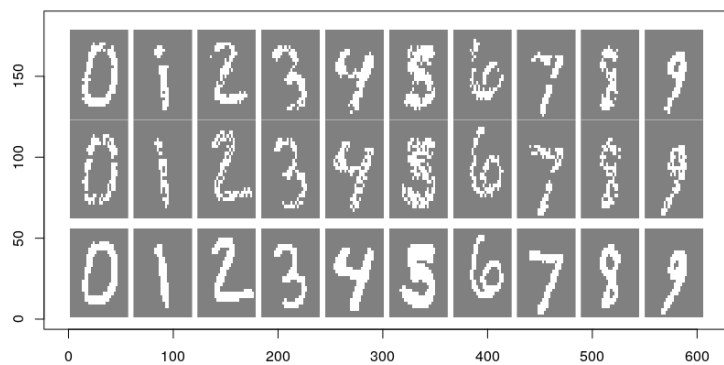


Figura 3: Resultados obtidos para o Autoencoder na execução 2

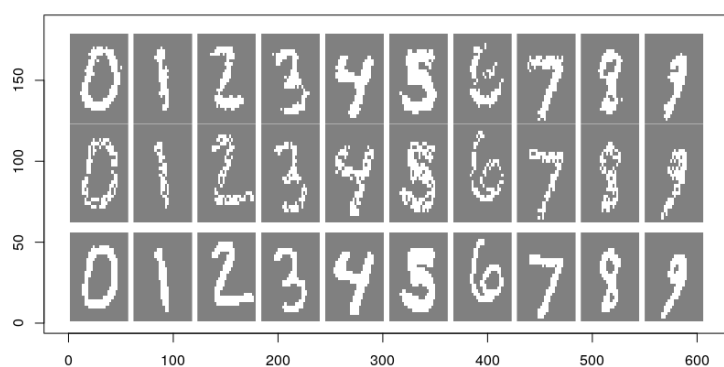


Figura 4: Resultados obtidos para o Denoising Autoencoder na execução 2

4.3 Execução 3

- Número de neurônios a mais = 90
- Taxa de Ruído = 0.3
- Imagens = 50

Com 90 neurônios a mais o desempenho para todas as redes cai um pouco. Nota-se que os dígitos não são reconstruídos com tanta precisão, como na Execução 2. De acordo com a análise sendo feita aqui, a qual é subjetiva e sujeita ao campo visual de quem observa as imagens resultantes pelas redes, conclui-se que, apenas aumentar o número de neurônios não melhora o desempenho, ao contrário, pode piorar. Como a Autoencoder e a Denoising Autoencoder são variações de Boltzmann então, de acordo com a teoria passada em sala de aula, que diz que Boltzmann não tem bom desempenho quando existem muitos neurônios, logo pode-se concluir que estas redes também terão o mesmo comportamento.

As Figuras 5, 6 e 7 expõem os resultados para o RBM, Autoencoder e Denoising Autoencoder, respectivamente, da execução 3.

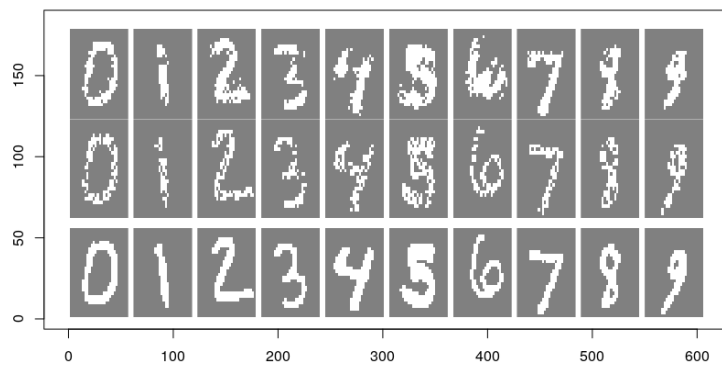


Figura 5: Resultados obtidos para a RBM na execução 3

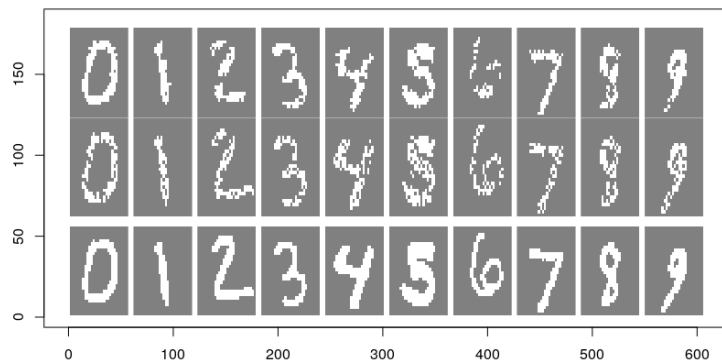


Figura 6: Resultados obtidos para o Autoencoder na execução 3

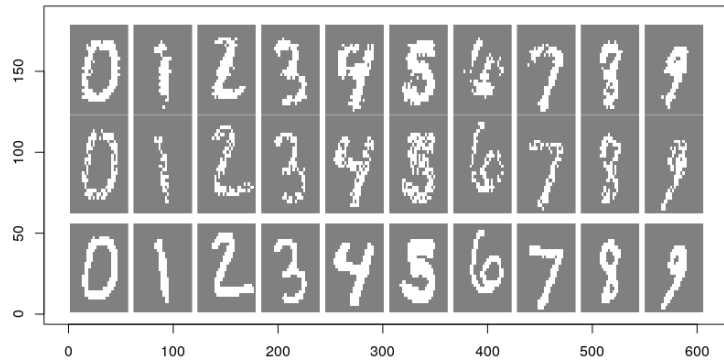


Figura 7: Resultados obtidos para o Denoising Autoencoder na execução 3

4.4 Execução 4

- Número de neurônios a mais = 60
- Taxa de Ruído = 0.2
- Imagens = 50

Nesta configuração muda-se apenas a taxa de ruído, o que nos permite observar que, mesmo mudando esse parâmetro, o resultado não muda muito. Observando as imagens resultantes geradas até o momento, é perceptível a dificuldade de todas as redes em reproduzir os números 5 e 6 e, nesta configuração, isto é ainda mais perceptível. Mais uma vez observamos que, independente das configurações dos parâmetros, a rede Autoencoder sempre gera melhor resultado, a RBM os piores e a Denoising Autoencoder sempre resultados entre ambos.

4.5 Execução 5

- Número de neurônios a mais = 60
- Taxa de Ruído = 0.3
- Imagens = 50

Esta configuração é igual à configuração número 2.

4.6 Execução 6

- Número de neurônios a mais = 60
- Taxa de Ruído = 0.5
- Imagens = 50

Nesta configuração, temos uma inversão no desempenho das redes. O resultado da Autoencoder é muito pior em relação aos resultados obtidos anteriormente e, inclusive muito pior em relação à Denoising Autoencoder e RBM. Diferentemente do que veio acontecendo até então,

a Denoising Autoencoder teve o melhor desempenho e a RBM um desempenho médio. Diante desta observação, pode-se concluir que o aumento da taxa de ruído em apenas 0.2 a mais que a anterior, afeta diretamente a reconstrução, piorando os resultados para a Autoencoder e RBM, e melhorando a Denoising Autoencoder.

As Figuras 8, 9 e 10 expõem os resultados para o RBM, Autoencoder e Denoising Autoencoder, respectivamente, da execução 6.

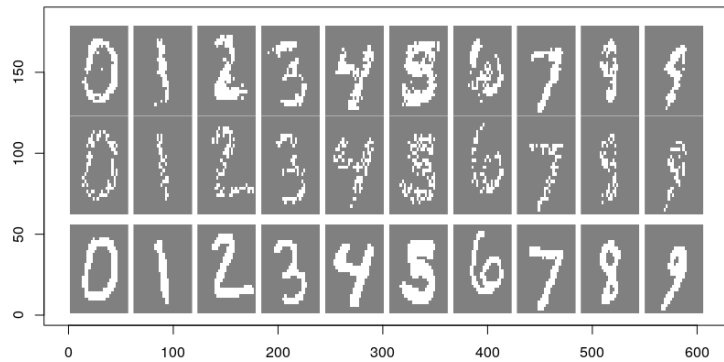


Figura 8: Resultados obtidos para a RBM na execução 6

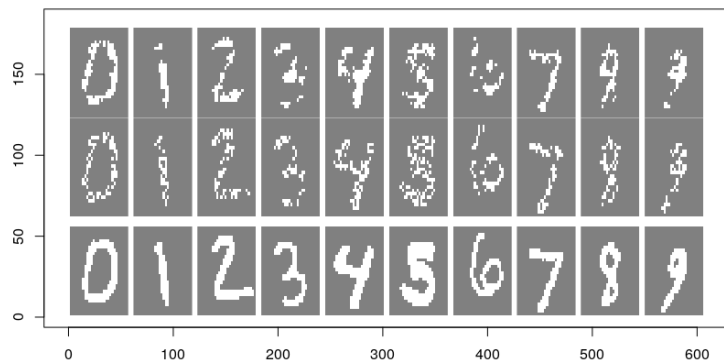


Figura 9: Resultados obtidos para o Autoencoder na execução 6

4.7 Execução 7

- Número de neurônios a mais = 60
- Taxa de Ruído = 0.3
- Imagens = 10

Nesta configuração a mudança está na quantidade de imagens passadas como parâmetros de entrada na rede. Até o momento estávamos passando sempre 50 imagens, o que até produzia bons resultados, caindo um pouco quando ou número de neurônios, ou a taxa de ruído,

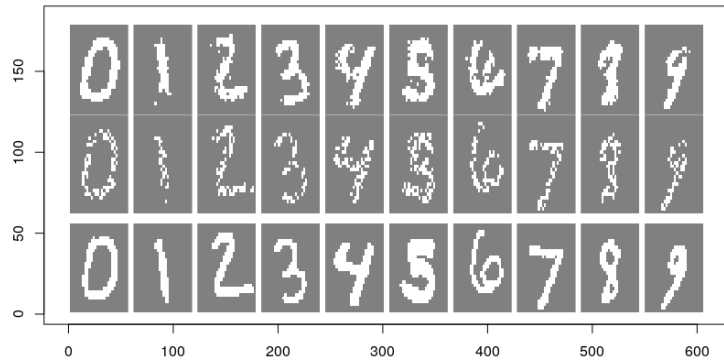


Figura 10: Resultados obtidos para o Denoising Autoencoder na execução 6

aumentava. Bem, alterar para um valor tão abaixo realmente piora e muito o desempenho da reconstrução. Nota-se claramente nas imagens resultantes a dificuldade das redes em reconstruir os dígitos originais. Portanto, um número muito baixo de imagens não é interessante, assim como uma taxa de ruído, ou grande número de neurônios também não. Concluimos então que é necessário encontrar um equilíbrio entre esses parametros para se obter o resultado desejado, que é a reconstrução a mais próxima possível da informação original.

As Figuras 11, 12 e 13 expõem os resultados para o RBM, Autoencoder e Denoising Autoencoder, respectivamente, da execução 7.

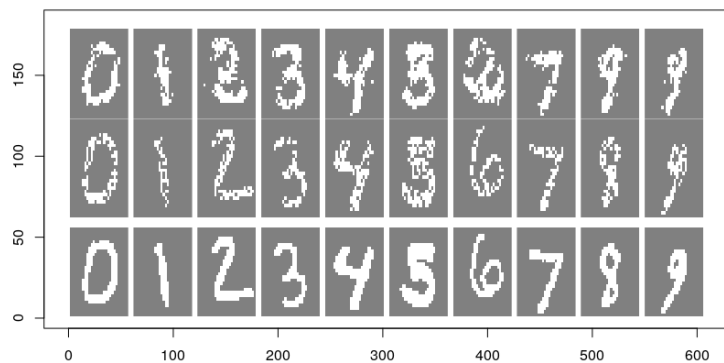


Figura 11: Resultados obtidos para a RBM na execução 7

4.8 Execução 8

- Número de neurônios a mais = 60
- Taxa de Ruído = 0.3
- Imagens = 50

Esta configuração é igual à configuração número 2.

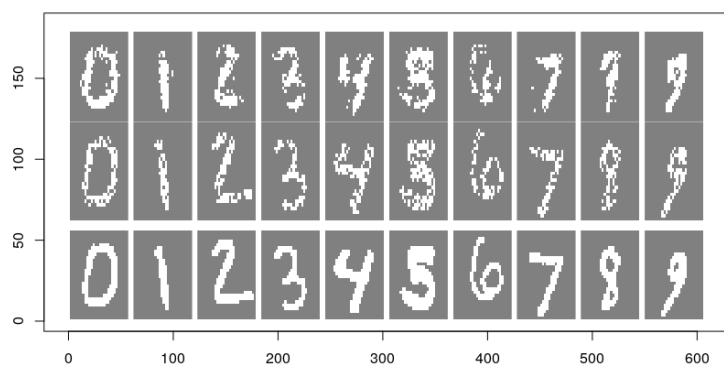


Figura 12: Resultados obtidos para o Autoencoder na execução 7

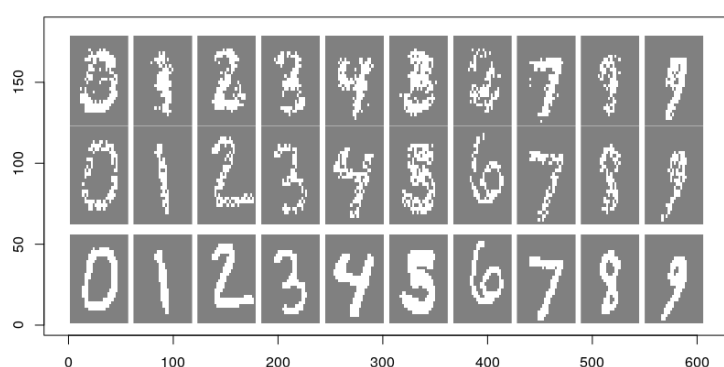


Figura 13: Resultados obtidos para o Denoising Autoencoder na execução 7

4.9 Execução 9

- Número de neurônios a mais = 60
- Taxa de Ruído = 0.3
- Imagens = 100

Nesta configuração dobramos o número de imagens quando comparada a Execução número 1, o que acabou por melhorar o resultado da reconstrução, sendo a rede Autoencoder novamente a rede com a reconstrução mais próxima do dígito original, a RBM com o pior resultado e a Denoising Autoencoder no meio termo.

As Figuras 14, 15 e 16 expõem os resultados para o RBM, Autoencoder e Denoising Autoencoder, respectivamente, da execução 9.

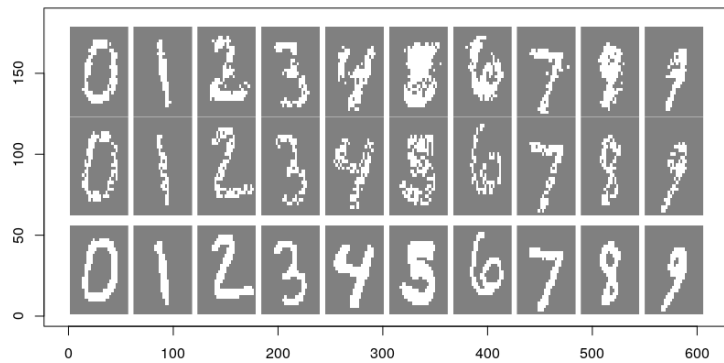


Figura 14: Resultados obtidos para a RBM na execução 9

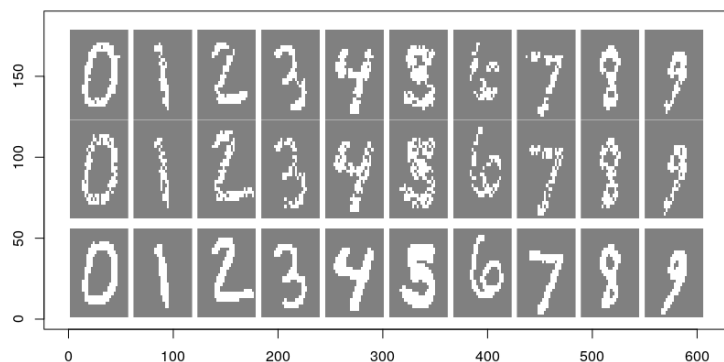


Figura 15: Resultados obtidos para o Autoencoder na execução 9

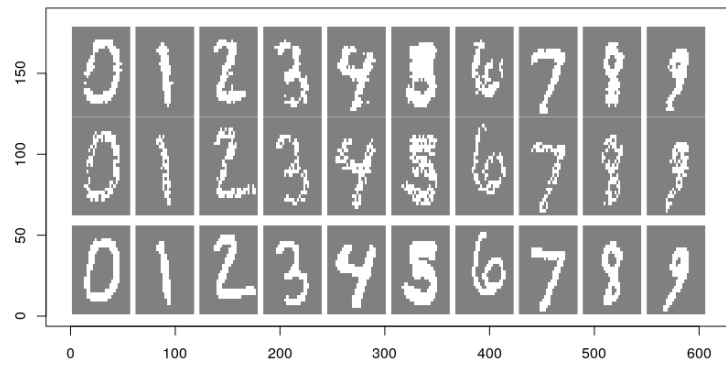


Figura 16: Resultados obtidos para o Denoising Autoencoder na execução 9

Referências

- [1] Z. L. Kovács. *Redes neurais artificiais*. Editora Livraria da Física, 2002.

L^AT_EX