

MLP vs CNN - AS1 - Curso de Redes Neurais

Análise Técnica Comparativa: MLP vs CNN no MNIST

Allan Ulisses Vigel Rodrigues

Redes Neurais (11100011158_20251_01)

1. Multilayer Perceptron (MLP)

1.1. Evolução Arquitetural e Hiperparâmetros

Teste	Camadas Ocultas	Ativação	Épocas	Otimizador	Batch size	Acurácia Validação (%)	Observações
1	1×(n=8)	sigmoid	5	RMSprop	512	28.0	Underfitting; capacidade insuficiente
2	1×(n=50)	sigmoid	5	RMSprop	512	79.0	Aumento de neurônios → ganho expressivo
4	1×(n=784)	softmax / cross-ent	2	RMSprop	512	86.22	Transição para softmax + cross-entropy
7	2×(784→360)	ReLU	5	RMSprop	512	92.24	ReLU reduz saturação, +6 p.p. de acurácia
9	2×(784→360)	ReLU	10	RMSprop	512	96.20	+5 épocas consolidam aprendizado
10	2×(784→360)	ReLU	30	RMSprop	512	96.72	Overfitting discreto; performance máxima

Figura 1. Curvas de treinamento e validação do MLP (Teste 10)

1.2. Análise Técnica (Causa e Efeito)

- **Capacidade de Representação:** ampliar de 8→784 neurônios (Teste 1→4) elevou a acurácia de ~27% a 86.2%, saindo do underfitting.
- **Profundidade vs Saturação:** inserir segunda camada (n2=360) com ReLU (Teste 4→7) mitigou gradientes saturados do sigmoid, acelerando convergência em +6 p.p. de acurácia.
- **Épocas de Treino:** estender de 5→30 épocas (Testes 7→10) refinou os pesos; a 10ª época (Teste 9) consolidou aprendizado e a 30ª (Teste 10) adicionou +0.5 p.p., mas aumentou val_loss (overfitting).
- **Otimizador & Batch:** RMSprop mostrou estabilidade; batches muito pequenos (<64) geram ruído, enquanto 512 equilibra estabilidade e velocidade.

🔑 **Conclusão MLP:** Melhor configuração (Teste 10):

- Arquitetura: Dense(784, ReLU) → Dense(360, ReLU) → Dense(10, Softmax)
- Épocas: 30 | Batch size: 512 | Otimizador: RMSprop
- Acurácia Validação: **96.72%**

1.3. Tabela completa:

Parâmetro	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Teste 6	Teste 7	Teste 8	Teste 9	Teste 10
n1	8	8	8	784	784	784	784	784	784	784
n2	desativado	desativado	desativado	desativado	desativado	360	360	784	360	360
n3	desativado	desativado	desativado	desativado	desativado	desativado	desativado	desativado	desativado	desativado
fa1	sigmoid	sigmoid	sigmoid	sigmoid	sigmoid	sigmoid	relu	relu	relu	relu
fa2	tanh	tanh	tanh	tanh	tanh	tanh	relu	relu	relu	relu
fas	relu	softmax	softmax	softmax	softmax	softmax	softmax	softmax	softmax	softmax
Otimizador	rmsprop	rmsprop	rmsprop	rmsprop	rmsprop	rmsprop	rmsprop	rmsprop	rmsprop	rmsprop
Função de perda (fp)	mean_absolute_error	mean_absolute_error	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy
Métrica	mean_squared_error	mean_squared_error	accuracy	accuracy	accuracy	accuracy	accuracy	accuracy	accuracy	accuracy
Épocas	2	2	2	2	2	2	2	2	10	30
Batch size	512	512	512	512	512	512	512	512	512	512
Treino (época final)	loss 0.0994 / MSE 0.0994	loss 0.1706 / MSE 0.1706	loss 0.0474 / accuracy 0.982	loss 0.6302 / accuracy 0.0	loss 0.7257 / accuracy 0.0	-	loss 0.3052 / accuracy 0.98	loss 0.3074 / accuracy 0.98	loss 0.0384 / accuracy 0.99	loss 0.00015 / accuracy 1.0
Validação (acurácia)	loss 0.0975 / MSE 0.0975	loss 0.1681 / MSE 0.1681	loss 0.0198 / accuracy 0.998	loss 0.5315 / accuracy 0.0	loss 0.6131 / accuracy 0.0	loss 0.7357 / accuracy 0.0	loss 0.2609 / accuracy 0.98	loss 0.2972 / accuracy 0.98	loss 0.1347 / accuracy 0.96	loss 0.1841 / accuracy 0.96

2. Convolutional Neural Network (CNN)

2.1. Evolução de Camadas e Hiperparâmetros

Teste	Convs (kernel)	Filtros C1→C2	Dense pós-Conv	Ativação	Épocas	% Treino	Otimizador + Perda	Acurácia Validação (%)	Observações
1	9×9 (tanh/sigmoid)	2→2	2	tanh	2	10%	Adadelta + hinge	~9.0	Underfitting extremo
3	3×3 (tanh/sigmoid)	4→8	10	tanh	5	10%	RMSprop + cross-ent	~25	Saturação ainda presente
6	3×3 (ReLU)	32→64	10	ReLU	10	70%	RMSprop + cross-ent	~97.8	Deep start melhora generalização
8	3×3 (ReLU)	32→64	256	ReLU	15	70%	RMSprop + cross-ent	98.61	Melhor balanço capacidade vs generalização

Figura 2. Curvas de treinamento e validação da CNN (Teste 8)

2.2. Análise Técnica (Causa e Efeito)

- **Kernel 3×3 vs 9×9:** kernels menores preservam detalhes locais, aumentando acurácia em ~15 p.p. ao migrar Teste 1→3.
- **Filtros Expandidos:** de 4→32 e 8→64 (Testes 3→6) ampliou a extração de features, elevando acurácia de ~25% a ~97.8%.
- **Proporção de Treino:** 10%→70% de treino (Teste 3→6) reduziu variação, melhorando generalização.
- **Batch & Épocas:** batch 256 + 15 épocas (Teste 8) refinou padrões finos, atingindo 98.61% com curvas estáveis.

🔑 **Conclusão CNN:** Melhor configuração (Teste 8):

- Arquitetura: [Conv3×3→ReLU→Pool]×2 → Flatten → Dense(256, ReLU) → Dense(10, Softmax)
- Épocas: 15 | Batch size: 256 | Otimizador: RMSprop
- Acurácia Validação: **98.61%**

2.3. Tabela completa:

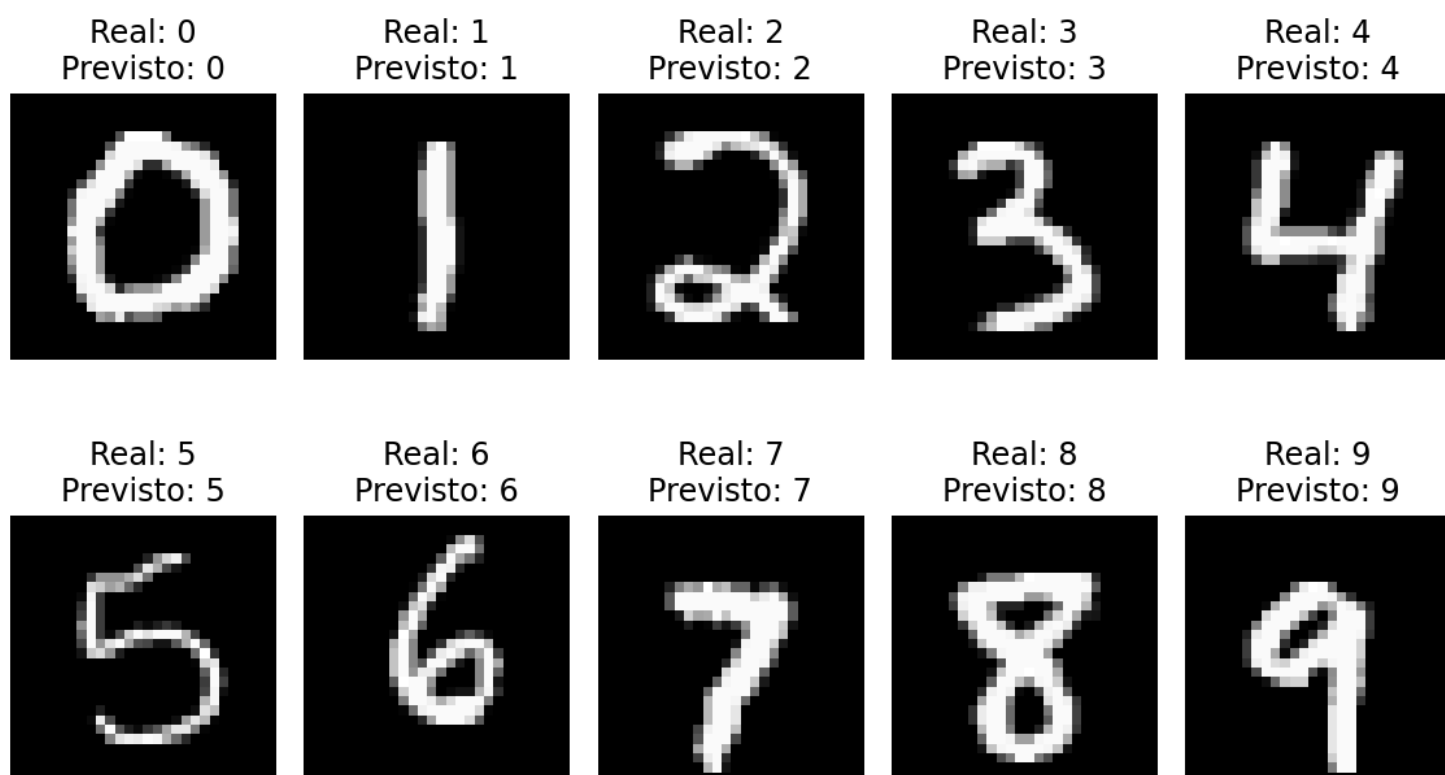
Parâmetro	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Teste 6	Teste 7	Teste 8
percentual_treinamento	0.10	0.10	0.10	0.70	0.70	0.70	0.70	0.70
m_C1	9	9	3	3	3	3	3	3
m_C2	9	9	9	9	9	3	3	3
n_C1	2	2	2	2	4	32	32	32
n_C2	2	2	2	2	8	64	64	64
n_D1	2	2	2	2	10	10	10	10
fa_C1	tanh	tanh	tanh	relu	relu	relu	relu	relu
fa_C2	tanh	tanh	tanh	relu	relu	relu	relu	relu
fa_D1	tanh	tanh	tanh	relu	relu	relu	relu	relu
fa_D2	sigmoid	sigmoid	sigmoid	softmax	softmax	softmax	softmax	softmax
otimizador	adadelta	rmsprop	rmsprop	rmsprop	rmsprop	rmsprop	rmsprop	rmsprop
função_perda	categorical_hinge	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy	categorical_crossentropy
metrica	MAE	accuracy	accuracy	accuracy	accuracy	accuracy	accuracy	accuracy
épocas	2	2	2	2	2	2	10	15
batch_size	1024	1024	1024	1024	1024	1024	512	256
Treino (última época)	loss 1.0288 / acc 0.0990	loss 2.2399 / acc 0.1962	loss 2.2549 / acc 0.2190	loss 2.1782 / acc 0.2327	loss 1.4690 / acc 0.5405	loss 0.5138 / acc 0.8680	loss 0.0744 / acc 0.9780	loss 0.0200 / acc 0.9938
Validação (última época)	val_loss 1.0293 / val_acc 0.0990	val_loss 2.2243 / val_acc 0.2286	val_loss 2.2407 / val_acc 0.2506	val_loss 2.1214 / val_acc 0.2506	val_loss 1.2498 / val_acc 0.6337	val_loss 0.3450 / val_acc 0.9110	val_loss 0.0827 / val_acc 0.9752	val_loss 0.0515 / val_acc 0.9861
Avaliação final	loss 1.0293 / acc 0.0959	loss 2.2243 / acc 0.2287	loss 2.2407 / acc 0.2525	loss 2.1214 / acc 0.2573	loss 1.2498 / acc 0.6337	loss 0.3450 / acc 0.9110	loss 0.0827 / acc 0.9752	loss 0.0515 / acc 0.9861

3. Comparativo Técnico MLP vs CNN

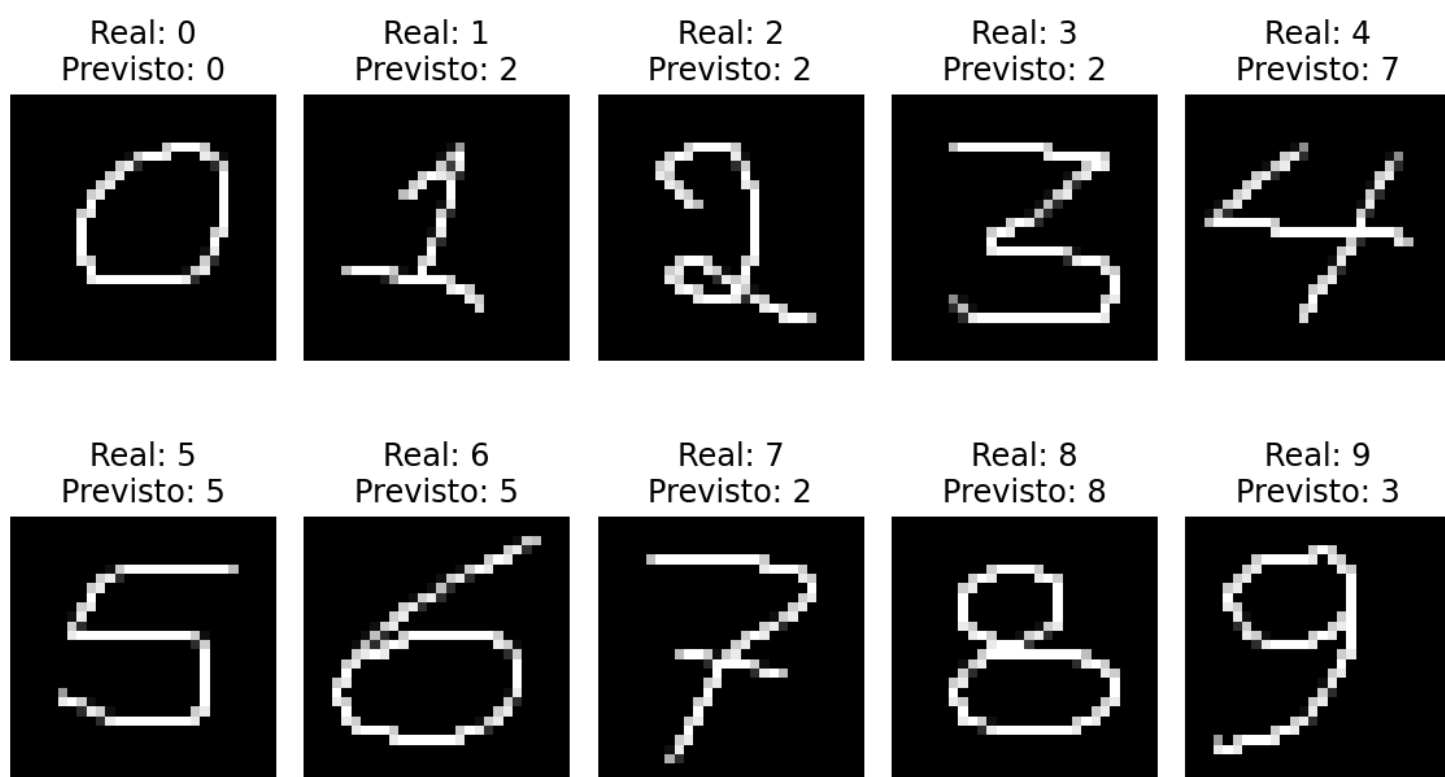
Métrica	MLP (Teste 10)	CNN (Teste 8)	Diferença	Interpretação Técnica
Acurácia Validação	96.72%	98.61%	+1.89 p.p.	CNN captura hierarquia espacial de pixels melhor
Épocas Convergência	30	15	−50%	CNN converge mais rápido por parâmetros locais
Generalização	<40% (Conj. 2)	~70% (Conj. 2)	+30 p.p.	CNN generaliza melhor em dados fora do treino

Insight: Convoluções 3×3 + pooling permitem à CNN extrair padrões locais antes de combinações densas, resultando em acurácia superior e melhor generalização com menos épocas, enquanto a MLP requer maior profundidade densa e mais ciclos de treino.

MLP Vt10 -> Conjunto 1:

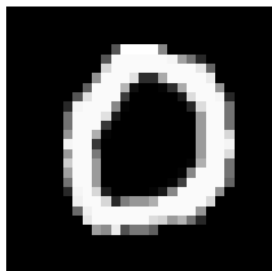


MLP Vt10 -> Conjunto 2:

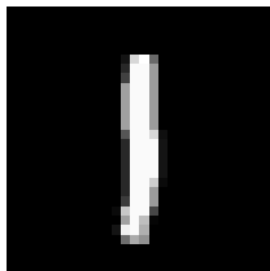


CNN Vt8 -> Conjunto 1:

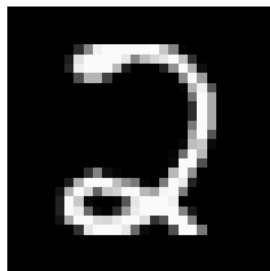
Real: 0
Previsto: 0



Real: 1
Previsto: 1



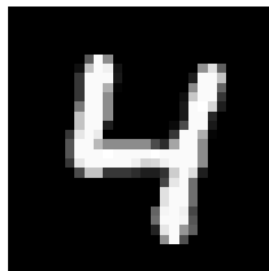
Real: 2
Previsto: 2



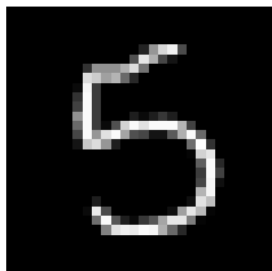
Real: 3
Previsto: 3



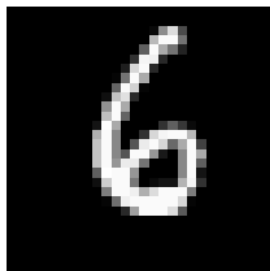
Real: 4
Previsto: 4



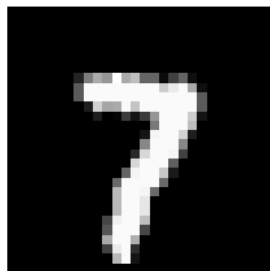
Real: 5
Previsto: 5



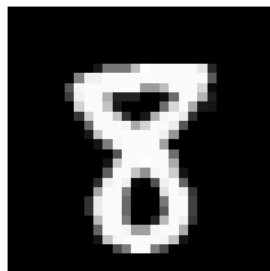
Real: 6
Previsto: 6



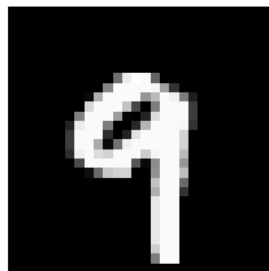
Real: 7
Previsto: 7



Real: 8
Previsto: 8

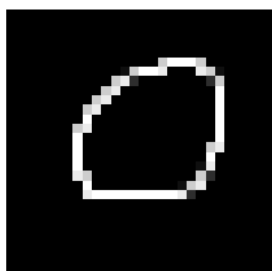


Real: 9
Previsto: 9

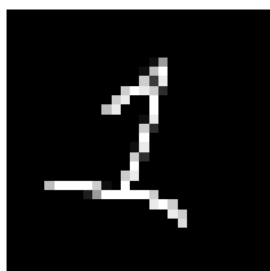


CNN Vt8 -> Conjunto 2:

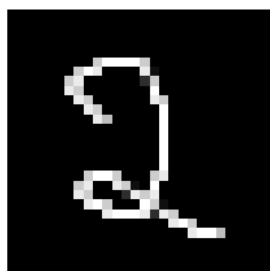
Real: 0
Previsto: 0



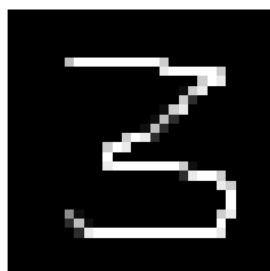
Real: 1
Previsto: 1



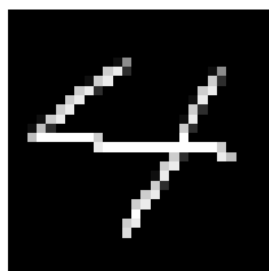
Real: 2
Previsto: 2



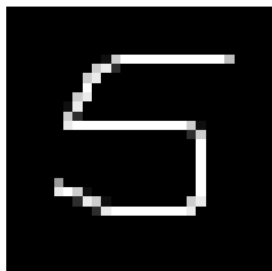
Real: 3
Previsto: 3



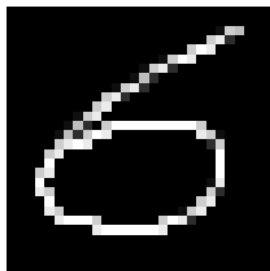
Real: 4
Previsto: 4



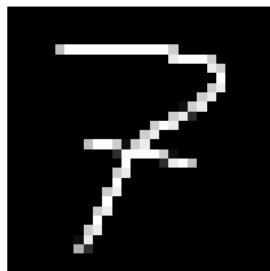
Real: 5
Previsto: 5



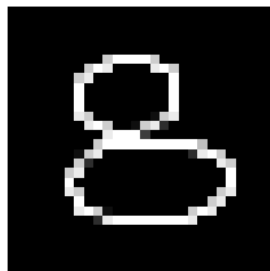
Real: 6
Previsto: 5



Real: 7
Previsto: 2



Real: 8
Previsto: 8



Real: 9
Previsto: 3

