

Uma comparação entre estratégias de investimento utilizando redes neurais, métodos estatísticos e *momentum*

Helcio Haruo Sasaki; Murilo Henrique Tank Fortunato

A inteligência artificial (IA) nasceu na metade do século XX e provocou duas “ondas” no mercado. A primeira começou em 1950, com Alan Turing^[1] como principal expoente, prevalecendo até meados de 1980, em meio aos avanços da ciência da computação. Um dos ícones dessa geração foi o desenvolvimento do jogo de xadrez para computadores. Uma segunda onda surgiu entre as décadas de 1990 e 2000, quando houve uma inversão na regra de programação clássica. Até então era necessário programar explicitamente todas as possibilidades para obter as respostas. Já no “machine learning”, o objetivo passa a ser treinar a máquina para o aprendizado, de forma que ela descubra a regra, conforme ilustrado na Figura 1.

Figura 1. Representação do modelo de “machine learning” antes e depois



Fonte: Elaborado pelos autores.

Atualmente já existem diversas aplicações práticas para produtos que utilizam o “machine learning”, como os assistentes pessoais, as ferramentas de marketing assertivo das redes sociais, os modelos de predição de diagnóstico aprimorados e, mais recentemente, os carros com direção autônoma (com destaque para o da empresa Tesla), o AlphaGo (IA da DeepMind que derrota humanos no jogo Go) e o ChatGPT (IA capaz de criar conteúdo), da OpenAI^[2].

Ultimamente o mercado financeiro passou a incorporar essas ferramentas, que têm enorme potencial para a descoberta de novas aplicações, a exemplo da utilização do “machine learning” para estimar o preço do café por meio da rede “convolutional neural network”^[3]. O objetivo deste trabalho foi implementar técnicas de “machine learning” para gerar *alfa* nos investimentos, ou seja, fazer uma gestão ativa obtendo retornos acima do “benchmark”. É importante mencionar que, em termos de acurácia, enquanto a máquina consegue obter níveis elevados de acerto em um modelo de classificação de imagens, no mercado financeiro o nível de predições corretas é mais modesto. Robert Mercer, um dos fundadores da gigante da gestão de investimentos Renaissance Technologies, uma vez disse: “estamos certos em 50,75% do tempo, mas estamos 100% certos em 50,75% das vezes”^[4].

Foi estruturada uma carteira cujos recursos foram igualmente distribuídos em quatro possíveis investimentos. A atualização foi mensal e sintetizou um histórico de 20 anos. Não foram considerados custos de corretagem, custos de execução, imposto de renda nem a possibilidade de posição vendida no ativo (aposta no retorno negativo dos ativos), uma vez que não é de fácil implementação para investidores pessoas físicas.

a) Dólar

O preço de referência do real para o dólar foi mensurado pela cota média de três fundos existentes (Santander Fundo Cambial, Itaú Fundo Cambial e Bradesco Fundo Cambial). Utilizar apenas a variação cambial não reflete corretamente a rentabilidade, pois existe uma remuneração sobre o caixa em dólares.

b) Ações do Ibovespa

Índice com as principais empresas negociadas na Bolsa de Valores, como: Petrobras, Vale, Itaú, Bradesco, Santander, Banco do Brasil, B3, WEG e Ambev.

c) Investimento com juros pré-fixados de dois anos

Taxa média interpolada para o prazo de dois anos, com base nos vértices de depósitos interfinanceiros negociados diariamente na Bolsa de Valores (B3). A escolha de um prazo genérico evitou a necessidade de se considerar o efeito de decaimento do prazo do papel. Nesta análise foi considerado o retorno de investimento com base no valor financeiro (PU):

$$PU_i = \frac{100.000}{(1 + taxa_i)^{du_i/252}} \quad (1)$$

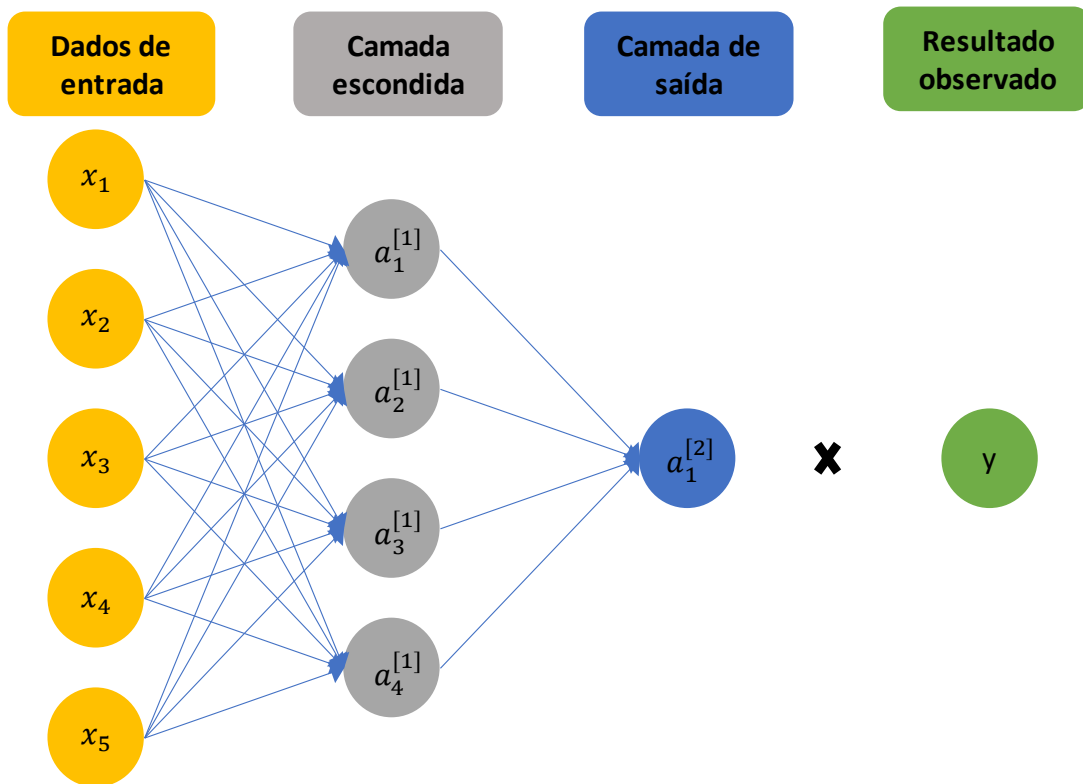
em que: PU_i é o valor presente dos 100 mil investidos; i representa a data de referência; du_i representa os dias úteis entre i e o prazo genérico de vencimento; e $taxa_i$ representa a taxa pré-fixada negociada no dia i .

d) Investimento com juros pré-fixados de cinco anos

Taxa média interpolada para o prazo de cinco anos, com base nos vértices de depósitos interfinanceiros negociados diariamente na Bolsa de Valores. Também foi considerado o retorno com base no valor financeiro (PU).

Linguagens de programação — como R, Java Script e, principalmente, Python — possuem diversos pacotes de redes neurais, como Keras e TensorFlow. Segundo Chollet^[5], essas ferramentas têm a vantagem de serem como blocos de montar, onde é possível encaixar camadas de funções de “deep learning” e utilizar o “pipeline” para ordenar o fluxo de informações dentro da estrutura. Porém, para ganhar sensibilidade do racional e dos resultados, foi criada uma rede neural do zero, detalhada na Figura 2^[6]:

Figura 2. Diagrama da rede neural proposta



Fonte: Elaborado pelos autores.

Neste exemplo existe cinco dados de entrada, uma camada escondida (ou camada intermediária) e uma camada de saída (ou camada de “output”) que contém o resultado estimado. A referência sobrescrita em colchetes ($[j]$) representa a camada de cada neurônio; neste caso, varia apenas entre 1 (camada “hidden layer”) e 2 (camada “output layer”). A referência subscrita “ i ” representa o neurônio contido em uma determinada camada. Nesse caso a camada escondida conta com quatro neurônios, e a camada de saída contém apenas um neurônio.

O neurônio $Z_i^{[j]}$ pode ser genericamente descrito como o resultado do “input” ponderado pelo peso $w^{[j]}$ (“weight”) somado ao viés $b^{[j]}$ (“bias”):

$$z_i^{[j]} = w^{[j]} x_i + b^{[j]} \quad (2)$$

Em cada uma das camadas, há uma função de ativação cujo objetivo é “ligar” ou “desligar” o neurônio. Essa etapa é importante para reduzir o risco de não convergência do erro, como descrito em Murray, Abrol e Tanner^[7] e Ovalle^[8], o que tornaria o modelo incapaz de encontrar uma solução.

Na camada escondida $a_i^{[1]}$ foi utilizada a função de ativação tangente hiperbólica \tanh , utilizando $w^{[1]}$ para ponderar os dados de entrada somado o viés $b^{[1]}$. Dessa forma o neurônio da camada intermediária tenta estimar o retorno do ativo entre alta (+1) ou queda (-1):

$$a_i^{[1]} = \tanh(w^{[1]} x_i + b^{[1]}) \quad (3)$$

$$a_i^{[1]} = \tanh(z^{[1]}) \quad (4)$$

Na camada de saída $a_i^{[2]}$ foi utilizada a função de ativação sigmoide σ . Desse modo, a camada de saída pondera o resultado da camada anterior $a_i^{[1]}$ pelo peso $w^{[2]}$ e adiciona o viés $b^{[2]}$, definindo assim se era interessante (+1) ou não (0) investir no ativo:

$$a_i^{[2]} = \sigma \left(w^{[2]} a_i^{[1]} + b^{[2]} \right) \quad (5)$$

$$a_i^{[2]} = \sigma \left(z^{[2]} \right) \quad (6)$$

Para o cálculo do erro, foi utilizada a equação “binary cross entropy”^[9]:

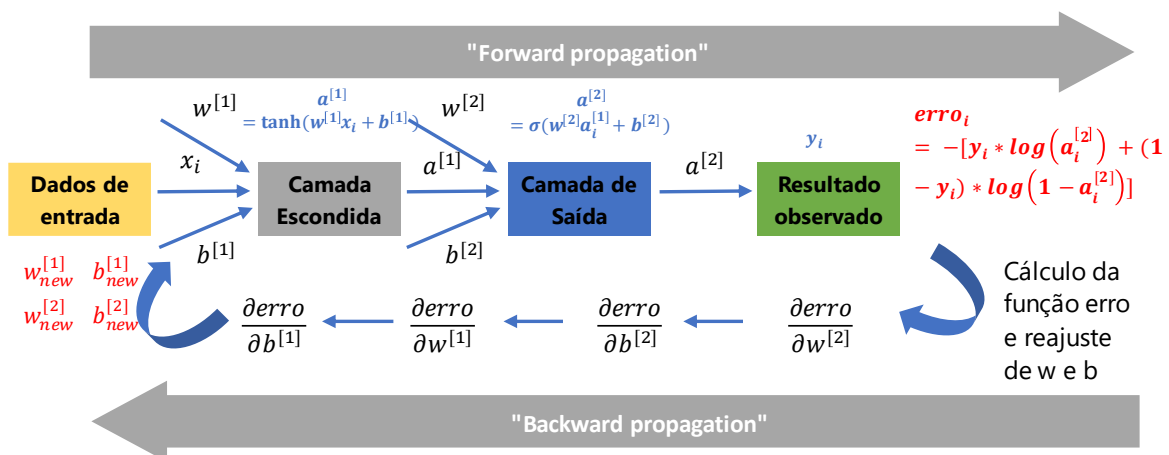
$$Erro\ total = -\frac{1}{m} \sum_{i=1}^m [y_i * \log(a_i^{[2]}) + (1 - y_i) * \log(1 - a_i^{[2]})] \quad (7)$$

em que: *erro total* representa a somatória da diferença entre o valor estimado e o valor realizado; *m* representa o tamanho da amostra; $a_i^{[2]}$ representa o valor estimado resultado do “output layer”; e Y_i representa a variável observada.

Inicialmente, na camada de entrada, os valores de $w^{[1]}$ e $b^{[1]}$ são valores aleatórios. O resultado da multiplicação dos “inputs” pelo peso $w^{[1]}$ é somado ao viés $b^{[1]}$, e o resultado final passa por uma função de ativação logística. A informação flui para a camada intermediária, em que o resultado anterior é multiplicado pelo novo peso $w^{[2]}$ e é somado ao viés $b^{[2]}$, os quais inicialmente também são valores aleatórios. Esse fluxo de informação é chamado de “forward propagation”.

A beleza da rede neural é que, com base na diferença entre o valor estimado e o valor realizado, é feita uma nova estimativa, com ajuste marginal dos parâmetros $w^{[j]}$ e $b^{[j]}$ para minimizar o desvio na próxima iteração. Isso se assemelha ao processo de aprendizado através de tentativa e erro, que nas redes neurais se chama “backward propagation”^[10].

Figura 3. Fluxo da informação dentro de uma rede neural



Fonte: Elaborado pelos autores.

Os parâmetros de correção descritos no “backward propagation” da Figura 3 foram obtidos através do método de otimização gradiente descendente, em que a correção do peso é definida por:

$$w_{new}^{[j]} = w_{old}^{[j]} - \text{alfa} * \frac{\partial \text{erro}}{\partial w^{[j]}} \quad (8)$$

em que: $w_{new}^{[j]}$ representa o novo peso; $w_{old}^{[j]}$ representa o peso anterior; *alfa* representa a taxa de aprendizado; e $\frac{\partial \text{erro}}{\partial w^{[j]}}$ representa a variação do erro dado uma variação do peso.

A correção do viés foi dada por:

$$b_{new}^{[j]} = b_{old}^{[j]} - \text{alfa} * \frac{\partial \text{erro}}{\partial b^{[j]}} \quad (9)$$

em que: $b_{new}^{[j]}$ representa o novo “bias”; $b_{old}^{[j]}$ representa o viés anterior; e $\frac{\partial \text{erro}}{\partial b^{[j]}}$ representa a variação do erro dada uma variação do viés.

Para o cálculo do “backward propagation”, foi resolvida a derivada parcial da sensibilidade do erro em relação ao peso da camada de “output”, $w^{[2]}$:

$$\frac{\partial \text{erro total}}{\partial w^{[2]}} = \frac{1}{m} (A^{[2]} - Y) A^{[1]t} \quad (10)$$

em que: $A^{[2]}$ representa o valor esperado na camada de saída em matriz; Y representa os resultados realizados em forma matricial; e $A^{[2]t}$ representa o resultado da camada intermediária em matriz transposta.

Na sequência foi calculada a derivada parcial da sensibilidade do erro em relação ao viés da camada de “output”, $b^{[2]}$, representada de forma matricial pela média do desvio:

$$\frac{\partial \text{erro total}}{\partial b^{[2]}} = \frac{1}{m} \sum_{i=1}^m (A^{[2]} - Y) \quad (11)$$

Até aqui foram calculadas as correções para o peso e para o viés da camada de saída. O próximo passo foi calcular o ajuste do peso e do viés para a camada escondida. A sensibilidade do erro em relação ao peso na camada intermediária, $w^{[1]}$, foi calculada da seguinte forma:

$$\frac{\partial \text{erro total}}{\partial w^{[1]}} = \frac{1}{m} (A^{[2]} - Y) W^{[2]t} * (1 - (A^{[1]})^2) X^t \quad (12)$$

em que: $W^{[2]t}$ representa os pesos da camada final em matriz transposta; e X^t representa os dados de “input” em matriz transposta.

Na sequência foi calculada a derivada parcial do erro em relação ao viés na camada intermediária, $b^{[1]}$:

$$\frac{\partial \text{erro total}}{\partial b^{[1]}} = \frac{1}{m} \sum_{i=1}^m (A^{[2]} - Y) W^{[2]t} (1 - (A^{[1]})^2) \quad (13)$$

em que: $W^{[2]t}$ representa os pesos da camada final em matriz transposta.

As fórmulas matriciais (10) e (11) da camada de saída e as equações (12) e (13) da camada intermediária foram codificadas em Python.

Todas as estratégias utilizaram 20 anos de histórico mensal, no período de janeiro de 2002 até dezembro de 2022.

a) Estratégia “long-only” (estratégia passiva)

Os recursos foram divididos igualmente entre quatro ativos: bolsa, dólar, juros de dois anos e juros de cinco anos, sem qualquer alteração durante o período de análise.

b) Momentum

Momentum representa a tendência de a alta dos preços se perpetuar nos próximos períodos, da mesma forma que a queda em geral é precedida por novas correções. Nesse sentido, os ganhadores do Nobel de economia de 2008, Fama e French^[12], afirmaram que *momentum* é um “premier market anomaly”, ou seja, uma estratégia que gera retornos acima da média. Trabalhos posteriores confirmaram essa anomalia, inclusive nos mercados de países emergentes^[11].

Antonacci^[13] foi além e propôs uma estratégia simples, mas eficaz, sobre o *momentum* dos ativos: se o retorno de um ativo for superior à taxa livre de risco, aplica-se nesse ativo para o próximo período. Caso o retorno seja inferior, aplica-se no ativo livre de risco.

c) Momentum de 21 períodos

Foi aplicada a mesma lógica da estratégia descrita no item anterior, mas utilizou-se uma média móvel dos retornos dos últimos 21 meses, prazo escolhido por ser comum em análises técnicas.

d) Regressão linear [“ordinary least squares (OLS)”]

Nessa estratégia foram calculados os coeficientes da regressão linear defasada em um período, conforme exemplo para o retorno do dólar:

$$\begin{aligned} USD_M = & \beta_1 USD_{M-1} + \beta_2 IBOV_{M-1} + \beta_3 PRE2y_{M-1} + \\ & + \beta_4 PRE5y_{M-1} + \beta_5 CDI_{M-1} \end{aligned} \quad (14)$$

em que: USD_M representa o retorno do dólar no período M ; USD_{M-1} é o retorno do dólar no período $M-1$; $IBOV_{M-1}$ é o retorno do Ibovespa no período $M-1$; $PRE2Y_{M-1}$ representa o retorno da aplicação em juros pré-fixados de dois anos no período $M-1$; $PRE5Y_{M-1}$ é o retorno da aplicação em juros pré-fixados de cinco anos no período $M-1$; CDI_{M-1} é o retorno do Certificado de Depósito Interbancário (CDI) no período $M-1$; e β_s representam os coeficientes da regressão.

Como detalhado adiante, o modelo de redes neurais utiliza 80% dos dados para treino. Para manter igualdade em relação ao tamanho da amostra, foi utilizado o mesmo corte de dados para estimar os coeficientes da regressão linear. Dessa forma, para um determinado mês “ M ”, 80% das observações anteriores (17 meses) foram utilizadas para estimar os coeficientes da regressão; os demais 20% das observações (quatro meses), como amostra para teste.

Se a estimativa do retorno do próximo período fosse positiva, os recursos seriam aplicados nesse ativo para o mês seguinte. Caso contrário, os recursos seriam aplicados no ativo livre de risco CDI.

e) Regressão logística

Da mesma forma que aconteceu com a regressão linear, foram estimados os coeficientes da regressão logística com base em uma janela móvel de 21 períodos: 80% das observações da amostra foram utilizadas para determinar os coeficientes da regressão, e os demais 20% como teste. Para definir se o retorno seguinte seria de alta (1) ou de baixa (0), foi utilizado o corte de 0,50 como limite.

f) Rede neural

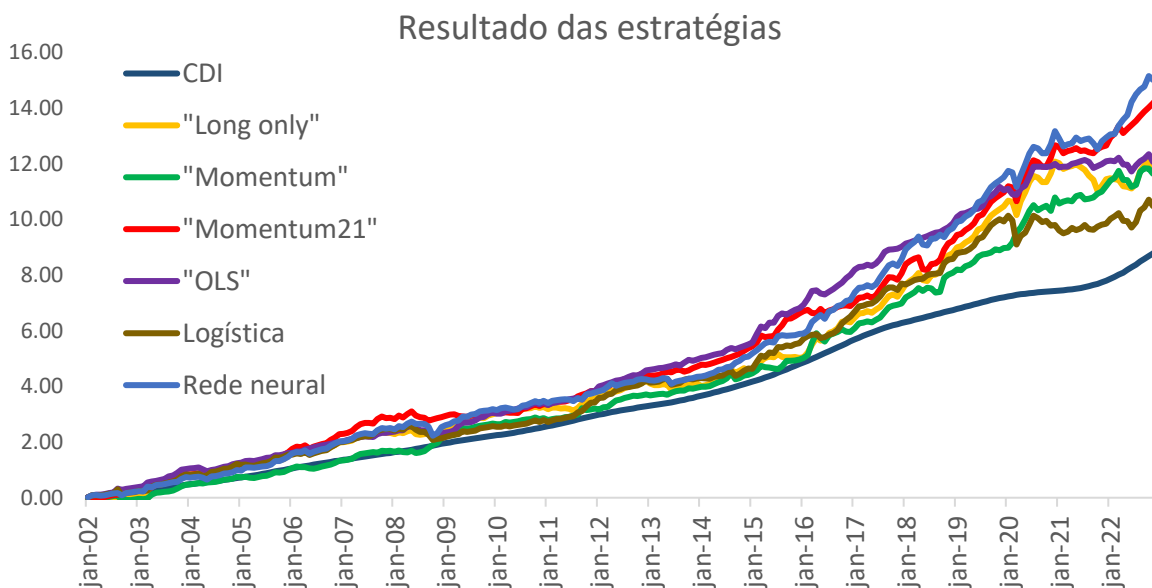
Foi implementada, conforme detalhado na Figura 2, uma camada escondida de quatro neurônios com ativação de tangente hiperbólica e um neurônio na camada de “output” com ativação logística.

A janela móvel continha o retorno dos últimos 21 meses; 80% das observações anteriores (17 meses) foram utilizadas para estimar os parâmetros de “weight” e “bias”, e os 20% restantes para o teste. A taxa de aprendizado foi de 0,50, com 1.000 iterações (“epochs”).

Caso o retorno esperado fosse positivo, o modelo aplicaria os recursos nesse ativo para o próximo mês. Caso o retorno esperado fosse negativo, o modelo aplicaria os recursos no CDI.

A Figura 4 demonstra o resultado acumulado de 20 anos das seis estratégias de investimento e o “benchmark”, que é o CDI acumulado no período. Na Tabela 1 constam os resultados detalhados por ano e estratégia.

Figura 4. Resultado mensal acumulado das 6 estratégias e o CDI acumulado



Fonte: Elaborado pelos autores.

Tabela 1. Resultado anual das diferentes estratégias

Retorno absoluto	CDI	"Long only"	"Momentum"	"Momentum21"	"OLS"	Logística	Rede neural
2002	19%	11%	-7%	24%	36%	20%	21%
2003	23%	74%	58%	49%	50%	52%	44%
2004	16%	13%	19%	16%	10%	21%	15%
2005	19%	15%	14%	23%	17%	15%	24%
2006	15%	20%	17%	25%	15%	18%	22%
2007	12%	11%	16%	18%	12%	16%	16%
2008	12%	4%	17%	1%	-2%	-10%	1%
2009	10%	15%	17%	5%	23%	14%	18%
2010	10%	6%	5%	7%	9%	6%	7%
2011	12%	7%	9%	11%	10%	17%	7%
2012	8%	14%	12%	12%	15%	18%	10%
2013	8%	-2%	5%	5%	7%	1%	2%
2014	11%	9%	9%	12%	9%	7%	14%
2015	13%	9%	10%	20%	20%	16%	13%
2016	14%	21%	17%	3%	15%	15%	18%
2017	10%	16%	15%	15%	11%	16%	18%
2018	6%	16%	14%	13%	8%	10%	12%
2019	6%	17%	10%	17%	11%	14%	17%
2020	3%	14%	18%	14%	8%	-1%	13%
2021	4%	-5%	4%	-0%	1%	0%	-2%
2022	12%	4%	5%	12%	1%	7%	16%
Desde o início	889%	1194%	1180%	1435%	1224%	1065%	1511%

Fonte: Elaborado pelos autores.

Com o objetivo de escolher a melhor estratégia, foram avaliados oito critérios: retorno total absoluto; “sharpe ratio” (retorno ajustado à volatilidade); “máximo drawdown” (perda máxima mensal); volatilidade (desvio padrão); número de anos negativos; número de meses negativos; “sortino ratio” (retorno ajustado à volatilidade de perdas); e número de anos acima do CDI.

Tabela 2. Medidas de avaliação das diferentes estratégias

	CDI	"Long only"	"Momentum"	"Momentum21"	"OLS"	Logística	Rede neural
Retorno total	889%	1194%	1180%	1435%	1224%	1065%	1511%
"Sharpe ratio"	-	0.56	0.67	1.36	0.98	0.38	1.28
Máximo drawdown"	-	-10%	-16%	-4%	-4%	-16%	-9%
Volatilidade	-	2.7%	2.2%	1.8%	1.7%	2.4%	2.2%
Anos negativos	-	2	1	1	1	2	1
Meses negativos	-	75	59	57	43	57	55
Sortino ratio"	-	0.66	0.60	1.53	1.14	0.32	1.25
Anos acima do CDI	-	7	9	9	8	9	11

Fonte: Elaborado pelos autores.

Para cada critério foram atribuídas notas de 1 a 6, sendo 6 a melhor e 1 a pior. Para os indicadores "sharpe ratio" e "máximo drawdown", houve ponderação de peso duplo, devido à sua importância para a avaliação de investimento.

Tabela 3. Ranqueamento da melhor estratégia (quanto maior a pontuação, melhor)

	Ponderação	"Long only"	"Momentum"	"Momentum21"	"OLS"	Logística	Rede neural
Retorno total	1x	3	2	5	4	1	6
"Sharpe ratio"	2x	4	6	12	8	2	10
Máximo drawdown"	2x	6	2	10	12	2	8
Volatilidade	1x	1	4	5	6	2	3
Anos negativos	1x	1	3	3	3	1	3
Meses negativos	1x	1	2	3	6	3	5
Sortino ratio"	1x	3	2	6	4	1	5
Anos acima do CDI	1x	1	3	3	2	3	6
Total de pontos		20	24	47	45	15	46

Fonte: Elaborado pelos autores.

Apesar da estratégia de redes neurais ter a melhor performance, com retorno de 1.511% em 20 anos, o modelo que teve a melhor pontuação foi o *momentum* com 21 períodos, com performance superior em itens como "sharpe ratio", "máximo drawdown", "sortino ratio" e volatilidade.

Na construção da estratégia de rede neural, notou-se variabilidade nos resultados a depender dos "inputs" utilizados, como taxa de aprendizado, quantidade de neurônios e número de simulações. Possivelmente pela falta de convergência do modelo ou "over-fitting" dos dados^[14]. A Tabela 5 contém a sensibilidade do resultado.

Tabela 5. Medidas de risco das diferentes estratégias de redes neurais

Variações dos Parâmetros da Rede Neural									
Núm. Neurônios	4	4	4	10	10	10	20	20	20
Tx. Aprendizado	0.10%	0.10%	0.50%	0.10%	0.10%	0.50%	0.10%	0.10%	0.50%
Núm. Simulações	1,000	5,000	1000	1,000	5,000	1,000	1,000	5,000	1,000
Retorno Total	1629%	1249%	1511%	1529%	1074%	1121%	1702%	1104%	1153%
Sharpe ratio	1.54	0.77	1.28	1.37	0.41	0.52	1.66	0.47	0.58
Máximo Drawdown	-9%	-13%	-9%	-9%	-16%	-13%	-9%	-16%	-13%
Volatilidade	2.1%	2.3%	2.2%	2.1%	2.4%	2.3%	2.1%	2.3%	2.3%
Anos negativos	2	2	1	1	1	1	3	1	2
Meses negativos	62	59	55	61	62	63	56	62	65
Sortino ratio	1.70	0.69	1.25	1.47	0.35	0.48	1.73	0.41	0.55
Anos acima do CDI	8	9	11	8	9	9	10	9	9

Fonte: Elaborado pelos autores.

O segundo ponto negativo das redes neurais foi a falta de sensibilidade de como as variáveis de entrada afetam o resultado. No caso de uma regressão, a sensibilidade de como as variáveis explicativas afetam o resultado pode ser observada pelos coeficientes da regressão, e, no caso da estratégia de *momentum*, a sensibilidade pode ser visualizada pela simples média dos últimos retornos.

De qualquer forma, todas as modelagens podem incorrer em erros causados pela falta de liquidez do ativo, surpresas na divulgação de indicadores, por exemplo, e inclusive pelo uso mais intenso de algoritmos que operam no mercado financeiro, pois o “algo trading” (“robôs”) pode aumentar o nível de ruído nos preços: se um ativo sobe indevidamente pela baixa liquidez, por exemplo, o computador pode notar a alta do preço e enviar ordem de compra, o que pode deslocar ainda mais o preço do ativo.

Porém, talvez o principal fator de distorção dos preços seja o viés comportamental dos investidores, que tende a apresentar oscilações entre os extremos de ganância e medo^[15], o que favoreceu a estratégia de *momentum*. A estratégia de investimento de rede neural com apenas uma camada intermediária e uma camada de saída, por sua vez, apresentou resultado bastante interessante e superou a rentabilidade absoluta das demais estratégias. No entanto, ao considerar outras métricas de desempenho, como retorno ajustado ao risco e volatilidade, a estratégia *momentum* com 21 períodos foi a alocação vencedora no período observado de 20 anos. Essa simples estratégia também possuiu duas vantagens em relação à rede neural:

- I. simplicidade do modelo de decisão, pois a escolha foi do ativo com rentabilidade superior à do CDI na média dos últimos 21 meses, diante da caixa preta dos resultados de “machine learning”, que, por não poderem ser reconstituídos, demandam confiança no “output” do modelo;
- II. facilidade na interpretação e previsibilidade do critério de decisão (um ativo que estiver subindo deve manter sua tendência), frente à falta de sensibilidade de como cada variável influencia a decisão do modelo de redes neurais.

É importante adicionar que os resultados das redes neurais apresentaram certa instabilidade nas simulações a depender dos parâmetros utilizados, o que gera insegurança com relação à estabilidade do modelo.

Por fim, como esse campo do conhecimento é recente, é importante olhar atentamente para os aperfeiçoamentos nos modelos de rede neural. Ao mesmo tempo, também é necessário humildade para reconhecer que pode haver limitações dos modelos de “machine learning”, em particular no mercado financeiro, uma vez que o comportamento dos investidores às vezes pode parecer mais errático do que racional, alternando entre momentos de euforia e de pânico.

Referências


- [1] Turing A. Computing Machinery and Intelligence. *MIND A Quarterly review of Psychology and Philosophy*. 1950; 59(236): 433-460. Disponível em: <https://psycnet.apa.org/doi/10.1093/mind/LIX.236.433>. Acesso em: 29 set. 2023.
- [2] Harari Y. ‘Sapiens’ author says AI is an alien threat that could wipe us out: ‘Instead of coming from outer space, it’s coming from California. *Fortune*. 2023. Disponível em: <https://fortune.com/2023/09/12/sapiens-author-yuval-noah-harari-ai-alien-threat-wipe-out-humanity-elon-musk-steve-wozniak-risk-cogx-festival/>. Acesso em: 29 set. 2023.
- [3] Santos J.A.A. Previsão do preço do café arábica: uma aplicação de redes neurais CNN-BLSTM. *Research, Society and Development*. 2022; 11(3): 1-11. DOI: <https://doi.org/10.33448/rsd-v11i3.26101>.
- [4] Zuckerman G. *O Homem que decifrou o Mercado*. Rio de Janeiro (RJ): Alta Books; 2020.
- [5] Chollet, F. *Deep Learning with Python*. Shelter Island (NY): Manning; 2018. 384 p.
- [6] Verma A. Shallow Neural Net from Scratch Using R. *RPubs*. 2020. Disponível em: <https://rpubs.com/theairbend3r/bin-classification-small-nnet-scratch-r>. Acesso em: 29 set. 2023.
- [7] Murray M.; Abrol V.; Tanner J. Activation Function Design For Deep Networks: Linearity And Effective Initialisation. *Applied and Computational Harmonic Analysis*. 2021; 59: 117-154. Disponível em: <https://arxiv.org/pdf/2105.07741.pdf>. Acesso em: 29 set. 2023.
- [8] Ovalle D. The purpose of activation functions in the neural network. *Medium*. 2020. Disponível em: <https://medium.com/@1087/the-purpose-of-activation-functions-in-the-neural-network-be08f81b7c32>. Acesso em: 29 set. 2023.
- [9] Yathish V. Loss Functions and Their Use In Neural Networks. *Medium*. 2022. Disponível em: <https://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9>. Acesso em: 29 set. 2023.
- [10] Patel J. Derivation of Backpropagation in Neural Network. *Youtube*. 2021. Disponível em: <https://www.youtube.com/watch?v=URJ9pP1aURo&t=190s>. Acesso em: 29 set. 2023.
- [11] Blitz D.; Hanauer M.; Vidojevic M. The idiosyncratic momentum anomaly. *International Review of Economics & Finance*. 2020; 69: 932-957. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S1059056020300927>. Acesso em: 29 set. 2023.
- [12] Fama, E.F.; French, K.R. Dissecting Anomalies. *The Journal of Finance*, 2008. Disponível em: <https://doi.org/10.1111/j.1540-6261.2008.01371>. Acesso em 29 set. 2023.
- [13] Antonacci G. *Dual Momentum Investing*. Nova York (NY): Mc Graw Hill Education; 2015. 256 p.
- [14] Dixon M.; Klabjan D.; Bang J.H. Classification-based financial markets prediction using deep neural networks. *Journal Algorithmic Finance*. 2017; 6(3-4): 67-77. Disponível em: <https://content.iospress.com/articles/algorithmic-finance/af176>. Acesso em: 29 set. 2023.
- [15] Kindleberger C.P.; Aliber, R.Z. *Manias, Panics, and Crashes: A History of Financial Crises*. Palgrave MacMillan 7th Edition. 2015 .

Como citar

Sasaki H.H.; Fortunato M.H.T. Uma comparação entre estratégias de investimento utilizando redes neurais, métodos estatísticos e *momentum*. Revista E&S. 2024; 5: e2023115.

Sobre os autores

Helcio Haruo Sasaki, Banco Santander, Avenida Juscelino Kubitschek, 2235 – Vila Olímpia – CEP 04543-011 – São Paulo/SP, Brasil.

Murilo Henrique Tank Fortunato , Instituto de Pesquisa e Educação Continuada em Economia e Gestão de Empresas – Pecege – Rua Cezira Giovanoni Moretti, 580 – Santa Rosa – CEP 13414-157 – Piracicaba/SP, Brasil.