Trabalho da Disciplina CMC 103 – Machine Learning

Crisitano Gurgel de Castro*

13 de novembro de 2024

Data de submissão: 13 de novembro de 2024.

1 Introdução

Na presente atividade, se propõe-se a conduzir uma análise de uma série de tarefas, cada uma focalizando diferentes parâmetros e hiperparâmetros associados às redes neurais artificiais. Com o objetivo principal de delinear a influência desses elementos técnicos sobre as propriedades comportamentais das redes neurais e obter elucidação de possíveis critérios e processos envolvidos na seleção desses parâmetros em aplicações práticas.

2 Tarefas Propostas

Para as presentes tarefas, uma rede neural foi dada como exemplo. Tal rede é treinada para um conjunto de dados (x_i, y_i) que refletem o comportamento de uma senóide $y_i = \sin(x_i) + r_i$. Um ruído r_i foi adicionado a cada medição de forma a refletir uma coleta de dados no mundo real com uma incerteza de medição associada

2.1 Tarefa 1 – Exploração de Funções de Ativação

> **Comando:** Nesta tarefa, você explorará o impacto de diferentes funções de ativação no desempenho da rede neural. A função de ativação controla como os neurônios transformam os dados de entrada, e diferentes funções podem influenciar a capacidade de aprendizado da rede.

A rede neural dada como exemplo utiliza a função Gelu como função de ativação nas camadas intermediárias. Vericamos o comportamento do erro de treinamento em relação ao número de épocas de treinamento (figura 1), bem como a comparação dos dados previstos pela rede dados em relação aos dados de treinamento na figura 2.

Para a presente tarefa modificamos a função de ativação das camadas intermediárias para sigmóide, obtendo os resultados mostrados nas figuras 3 e 4. Observamos que a com

Figura 1 – Gelu: Treinamento

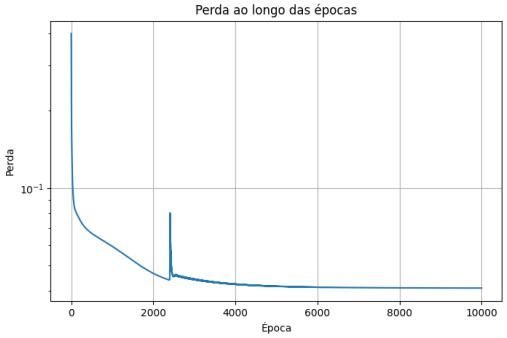


Figura 2 – Gelu: predições

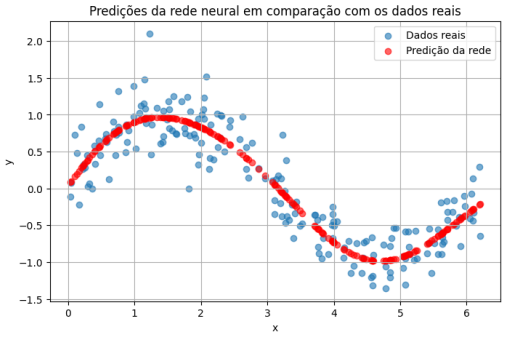


Figura 3 – Sigmóide: Treinamento

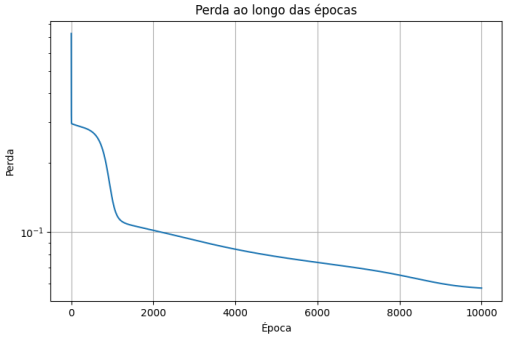
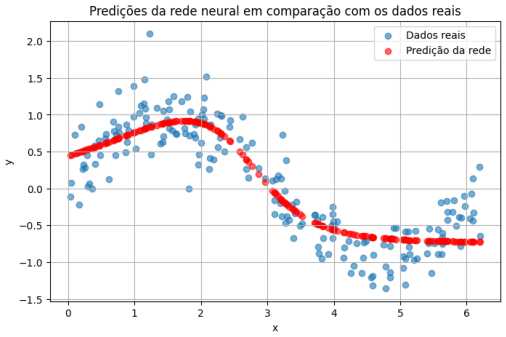


Figura 4 – Sigmóide: predições



a função sigmóide ($\sigma(\cdot)$)a rede não é capaz de acompanhar as não linearidades inerentes a função seno de maneira tão precisa quando a Gelu.

Em uma segundo etapa, a função de ativação das camadas intermediárias foi modificada para a função Relu. Obtivemos os seguintes resultados da evolução das perdas durante o treinamento e do teste de predição da rede neural treinada mostrados nas figs. 5 e 6. Observamos que com essa função de ativação obtivemos o gráfico de perda ao longo das épocas mais ruidoso. O que pode indicar uma inicialização de parâmetros e/ou taxa de aprendizada que pode ser melhorada para este tipo de rede neural. Em especial a inicialização do *bias* requer uma atenção especial nesse caso, visto que para a função Relu é adequado um bias inicial maior que 0. Outro ponto interessante que pode ser observado no predição da rede neural é que o modelo treinado não parece replicar uma curva continuamente diferenciável, mas tenta replicar a senóide através de trechos de funções afins. Essa observação vai ao encontro da característica da função Relu a qual não é continuamente diferenciável.

Por fim, comparamos na tabela 1 os erros de treinamento para a época 9900 de cada uma das redes neurais com diferentes funções de ativação. Observamos que embora a Relu não consiga replicar o comportamento da senóide, ela possui ao final o menor erro de treinamento. No entanto, esta rede neural pode ter problemas com generalização.

2.2 Tarefa 02 – Variação do Número de Épocas

Comando: Objetivo. Analisar como a quantidade de épocas de treinamento influencia o desempenho da rede neural na aproximação da

*Inmetro, (21) 2679 9799, cgcastro at inmetro gov br

Figura 5 – Relu: Treinamento

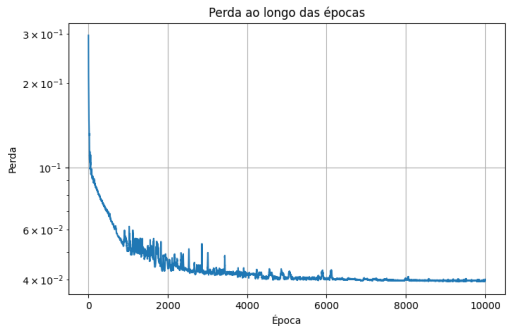


Figura 6 – Relu: predições

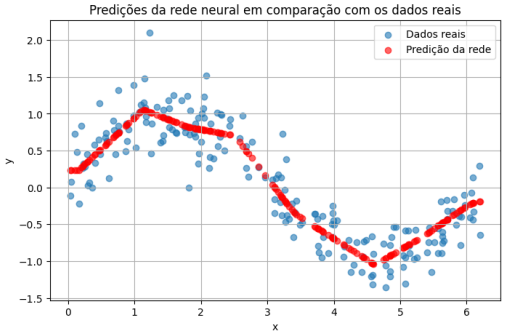


Tabela 1 – Comparação das perdas de treinamento

função	época	perda
Gelu	9900	0.04087
Sigmóide	9900	0.05731
Relu	9900	0.03939

função seno. Observaremos como diferentes números de épocas afetam a convergência da perda e a qualidade das predições da rede.

No presente treinamento, treinamos 4 redes neurais distintas. Todas tem a mesma características, mesmo modelo, mesmas funções de ativação nas camadas intermediárias (gelu) e mesma taxa de aprendizado. No entanto cada uma delas foi treinada com um número diferente de épocas. Vemos os resultados nas figs. 7 à 10 a seguir. Na fig. 11 tem-se as curvas de erros durante o treinamento para as redes neurais. Na tabela 2 vemos as perdas para um mesmo conjunto de dados de teste referentes às diferentes redes neurais.

Tabela 2 – Perda de teste para as diferentes redes neurais

Rede Neural	Perda com dados de teste
1k épocas	0.06468
5k épocas	0.04504
10k épocas	0.04528
20k épocas	0.04492

Após 5000 épocas de treinamento, foi observado que as redes neurais apresentam erros muito próximos uns dos outros. Isso indica que, a partir desse ponto, aumentar ainda mais o número de épocas de treinamento não traz melhorias significativas na capacidade preditiva das redes. Este achado é importante, pois sugere que as redes alcançaram um nível de desempenho máximo e que o treinamento adicional pode não ser benéfico.

Um caso especial que chama a atenção é a rede neural treinada com apenas 1000 épocas. Esta rede exibe um erro de previsão significativamente maior do que as outras redes, indicando um possível caso de *underfitting*. *Underfitting* ocorre quando um modelo é muito simples para capturar os padrões complexos nos dados, resultando em um desempenho de previsão inferior.

Uma observação secundária interessante é a comparação entre a rede treinada com a função de ativação Gelu por 1000 épocas e a rede treinada com a função de ativação

Figura 7 – Treinando com 1k épocas.

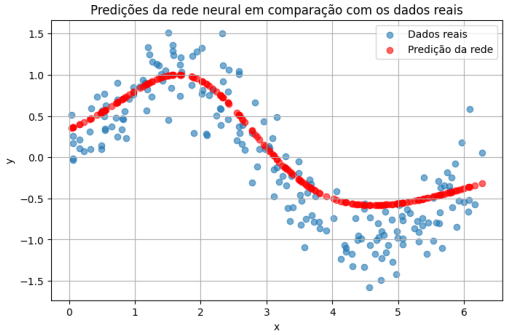


Figura 8 – Treinando com 5k épocas.

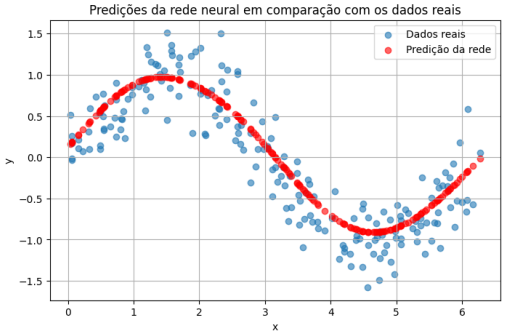


Figura 9 – Treinando com 10k épocas.

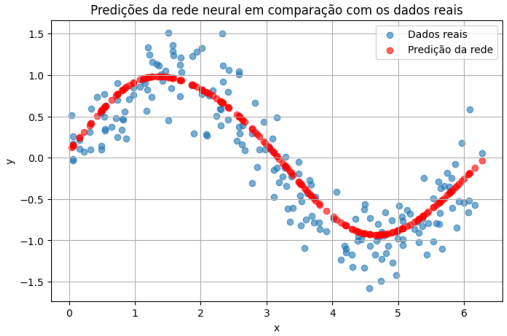


Figura 10 – Treinando com 20k épocas.

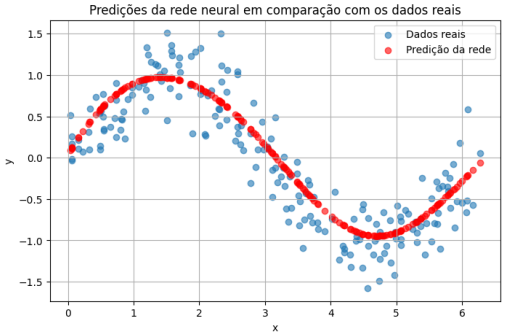
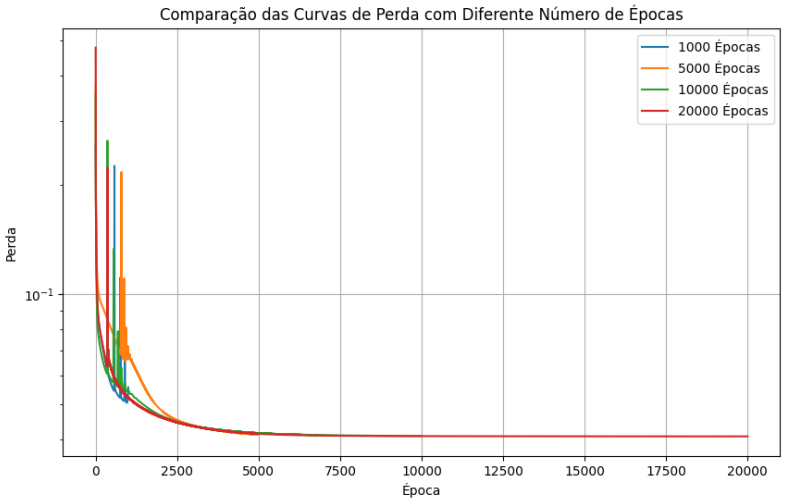


Figura 11 – Curvas de perdas durante o treinamento para as diferentes redes



sigmóide. Parece que ambas as redes têm capacidades preditivas semelhantes, apesar da diferença na função de ativação e no número de épocas de treinamento. Isso pode sugerir que, neste caso específico, a função de ativação Gelu com menos épocas de treinamento pode alcançar um desempenho comparável ao sigmóide (fig. 4).

Em resumo, esta análise destaca a importância de monitorar o desempenho das redes neurais em diferentes pontos do treinamento. Mostra que o *underfitting* pode ocorrer com menos épocas de treinamento e que diferentes funções de ativação podem ter impactos variados no desempenho do modelo. Os resultados também sugerem que, após um certo

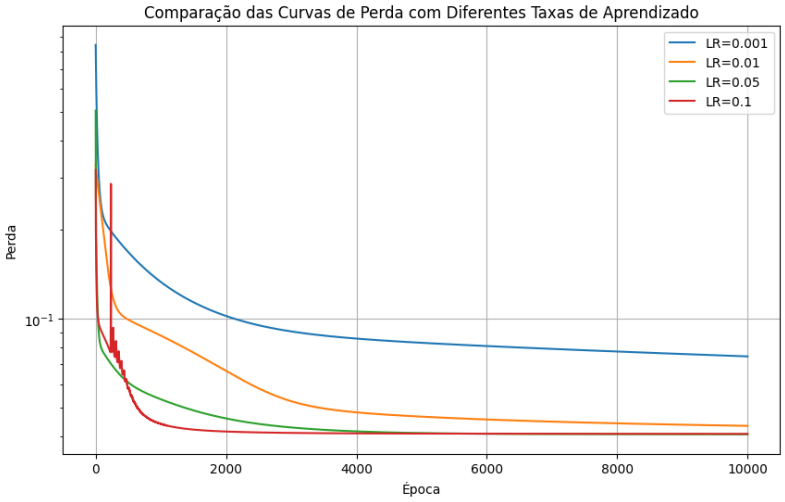
número de épocas, a melhoria no desempenho da rede neural pode ser marginal, indicando que a otimização adicional pode não ser necessária.

2.3 Tarefa 03 – Influência da Taxa de Aprendizado

> **Comando:** Objetivo. Investigar como diferentes taxas de aprendizado (`learning_rate`) afetam a convergência e o desempenho da rede neural. Analisaremos como a velocidade e a estabilidade do treinamento variam com a alteração dessa taxa.

Nessa atividade treinamos uma mesma rede neural com os mesmos dados de testes, variando apenas a taxa de aprendizado (α) e verificando o seu comportamento durante o treinamento e a capacidade de previsão da rede. Treinamos as redes para $\alpha = 0,001|0,01|0,05|0,1$. A evolução do erro durante o treinamento para as redes é mostrada na fig. 12. As capacidades de previsão das redes são mostradas nas figs. 13 à 16.

Figura 12 – Curvas de perda durante treinamento (modificando α)



Ao analisar os resultados com uma taxa de aprendizagem de 0,001, observa-se que a convergência é significativamente mais lenta. O modelo parece ficar longe dos outros em termos de desempenho. Isso sugere que o modelo pode ter caído em um mínimo local, ou que, são necessárias muito mais épocas de treinamento para atingir a convergência.

Por outro lado, uma taxa de aprendizagem de 0,05 parece ser um bom equilíbrio. Oferece uma convergência mais suave e estável, evitando oscilações extremas, e resulta em um bom erro de teste no final do treinamento. Este α pode ser uma escolha ideal, pois encontra um compromisso entre a velocidade de convergência e a precisão do modelo.

Quando a taxa de aprendizagem é aumentada para 0,1, o gráfico da curva de aprendizado parece mais ruidoso e volátil. Isso indica que o modelo pode estar enfrentando dificuldades para encontrar o mínimo do gradiente nas primeiras épocas do treinamento. Taxas de aprendizagem mais altas podem fazer com que o modelo pule o mínimo global e resultem em um comportamento mais caótico durante o processo de otimização.

Em resumo, a escolha da taxa de aprendizagem é crucial para o treinamento de modelos de aprendizado de máquina. Uma taxa de aprendizagem muito baixa pode resultar em convergência lenta, enquanto uma taxa de aprendizagem muito alta pode levar a

Figura 13 – $\alpha = 0,001$

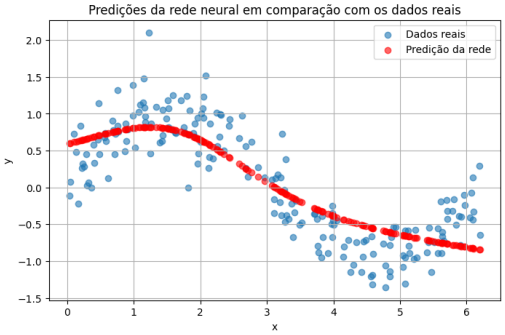


Figura 14 – $\alpha = 0,01$

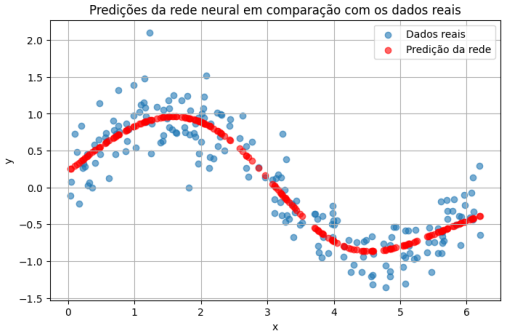


Figura 15 – $\alpha = 0,05$

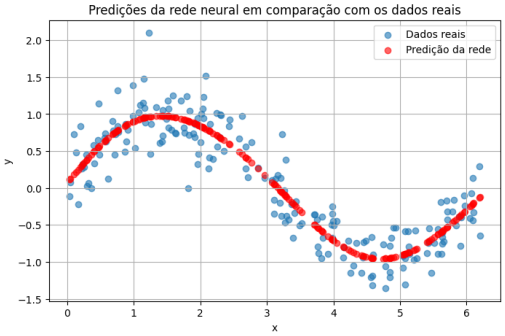
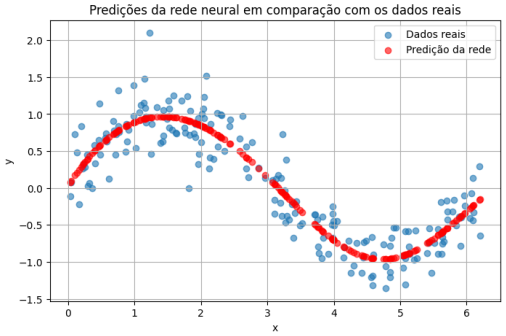


Figura 16 – $\alpha = 0,1$



um comportamento instável. Uma taxa de aprendizagem de 0,05 parece ser uma escolha adequada para este cenário específico, proporcionando uma convergência suave e um bom desempenho final.

2.4 Tarefa 04 – Ajuste do Número de Neurônios por Camada

Comando: Objetivo: Avaliar como a quantidade de neurônios em cada camada oculta influencia a capacidade de aprendizagem e a performance da rede neural. Exploraremos diferentes arquiteturas para entender o impacto da complexidade da rede.

Foram realizados experimentos de treinamento com quatro diferentes configurações de neurônios nas camadas ocultas, mantendo o número de épocas constante em 10000. A primeira configuração com $[1, 5, 5, 5, 1]$ neurônios em cada camada resultou em uma perda de treinamento de $\approx 0,04058$. A segunda configuração com $[1, 10, 10, 10, 1]$ neurônios produziu uma perda de ≈ 0.04076. A terceira configuração com $[1, 20, 20, 20, 1]$ neurônios resultou em uma perda de ≈ 0.04071, e a quarta configuração com $[1, 50, 50, 50, 1]$ neurônios teve uma perda de ≈ 0.04078. As curvas de treinamento podem ser vistas na fig. 17

O efeito no tempo de treinamento para cada arquitetura pode ser visto na fig. 18. É importante notar que, conforme veremos mais adiante e com base na quantidade de dados disponível, não é necessário ter camadas muito densas com muitos neurônios.

As predições das redes são mostradas nas figs. 19 à 22 A configuração mais simples, com menos neurônios, apresenta uma boa convergência e uma curva de perda suave, sem muitas oscilações. Isso implica que, para este conjunto de dados específico, uma arquitetura de rede neural menos complexa pode ser suficiente para obter bons resultados.

Figura 17 – Curvas de aprendizados para diferentes configurações da rede

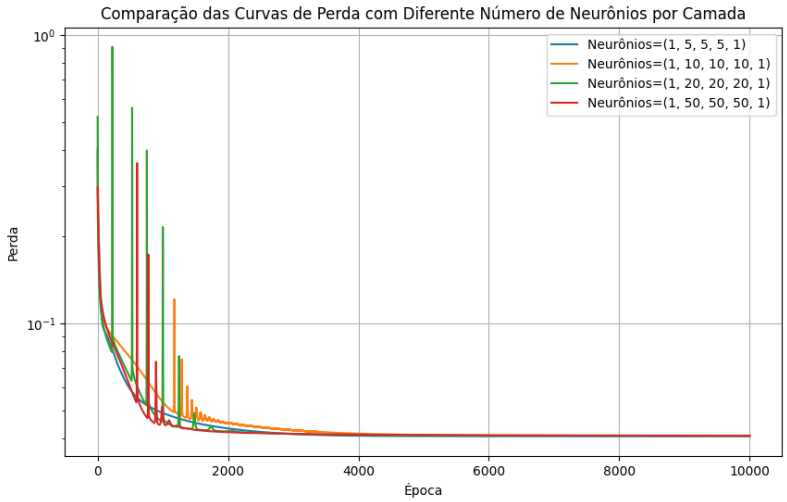
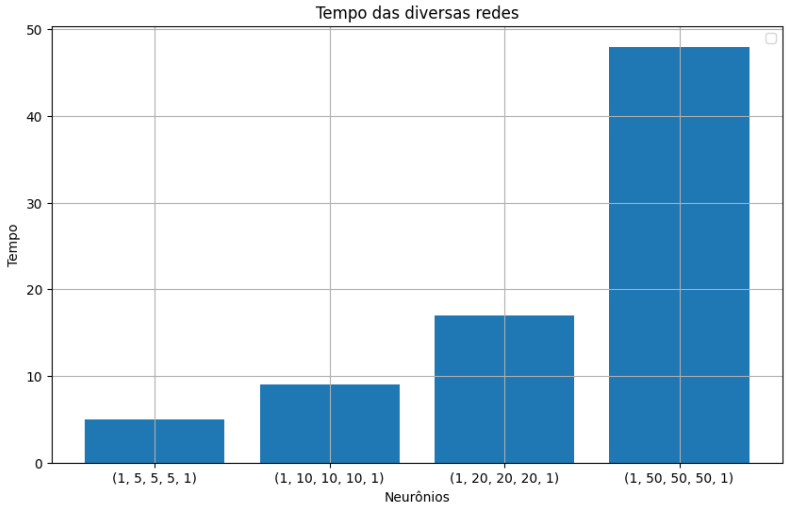


Figura 18 – Tempo de treinamento de acordo com a arquitetura da rede



Em resumo, esta análise destaca a importância de considerar a complexidade da arquitetura da rede neural em relação à quantidade de dados disponíveis. Em alguns casos, uma rede neural mais simples pode convergir bem e fornecer resultados comparáveis a redes mais profundas e complexas. A escolha da arquitetura ideal depende dos dados específicos e dos objetivos do problema em questão.

2.5 Tarefa 5: Introdução de Ruído nos Dados de Treinamento

Comando: Objetivo: Investigar como diferentes níveis de ruído nos dados de treinamento afetam a performance e a generalização da rede neural. Avaliaremos a robustez do modelo frente a dados ruidosos.

Foram realizados experimentos de treinamento com quatro níveis diferentes de ruído adicionados aos dados: $\sigma = 0.0$, $\sigma = 0.1$, $\sigma = 0.3$ e $\sigma = 0.5$. O gráfico de treinamento para as diferentes redes é apresentada na fig. 23

Figura 19 – Configuração $[1, 5, 5, 5, 1]$

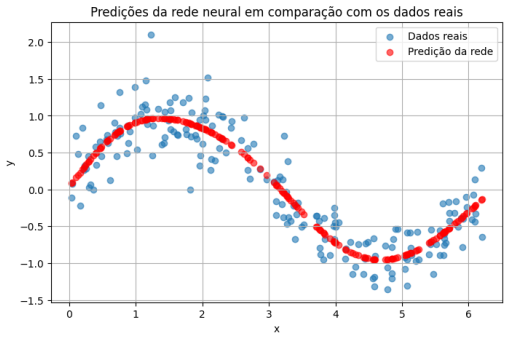


Figura 20 – Configuração $[1, 10, 10, 10, 1]$

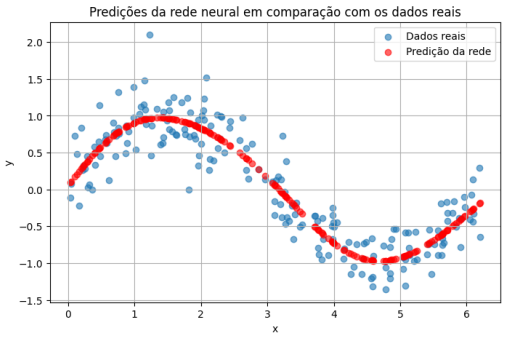


Figura 21 – Configuração $[1, 20, 20, 20, 1]$

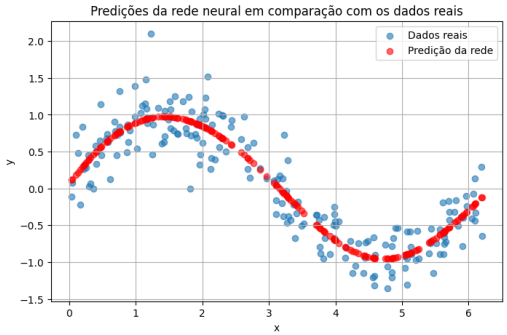


Figura 22 – Configuração $[1, 50, 50, 50, 1]$

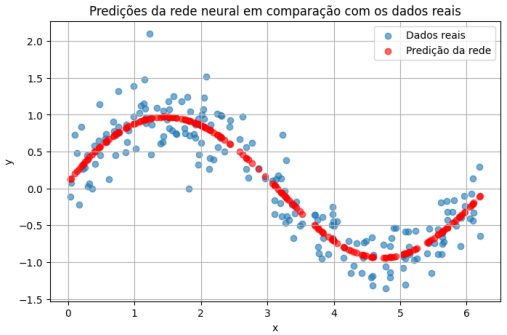
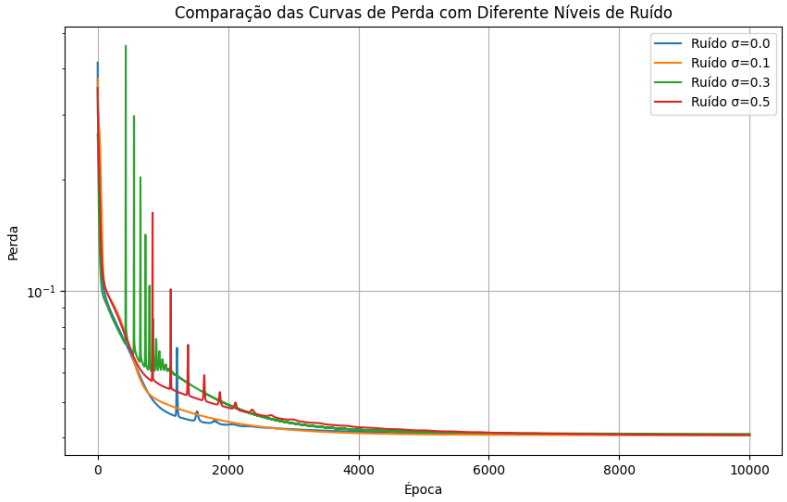


Figura 23 – Curvas de aprendizado para diferentes níveis de ruído dos dados de entrada



Para $\sigma = 0{,}0$, que representa dados sem ruído, os gráficos de perda mostram uma convergência suave e rápida, indicando que o modelo está aprendendo bem.

Para as redes com níveis de ruídos maiores, a convergência é ligeiramente mais lenta e a perda é um pouco maior em comparação com os dados sem ruído. Isso é esperado, pois o ruído adiciona complexidade aos dados, tornando a tarefa de aprendizagem mais desafiadora.

Com $\sigma = 0,3$, a convergência torna-se mais instável, especialmente nas primeiras

épocas do treinamento. A perda oscila mais e leva mais tempo para estabilizar. Isso sugere que o modelo está encontrando dificuldades para aprender os padrões subjacentes nos dados devido ao alto nível de ruído.

Para $\sigma = 0,5$, é notável que a convergência parece mais estável do que $\sigma = 0,3$ nas primeiras épocas, sugerindo que o modelo está encontrando maneiras de lidar com o ruído mais intenso. No entanto, como decorrer das épocas a taxa de erro volta a ficar maior que com o $\sigma = 0,3$ como era esperado.

A análise geral dos resultados revela uma relação interessante entre o nível de ruído e a convergência do modelo. Quanto maior o ruído, mais demorada é a convergência, o que é intuitivo, pois o modelo precisa lidar com mais incerteza nos dados. Apesar do ruído introduzido, pode-se ver nas figs. 24 à 27 que a predição das redes são adequadas.

Figura 24 – $\sigma = 0.0$

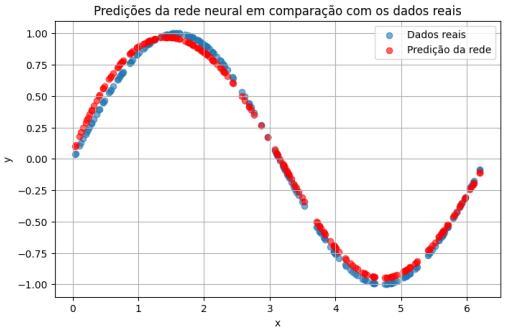


Figura 25 – $\sigma = 0.1$

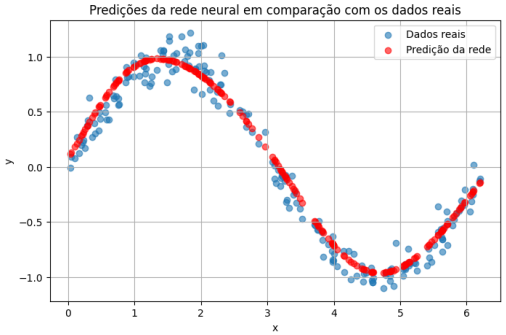


Figura 26 – $\sigma = 0.3$

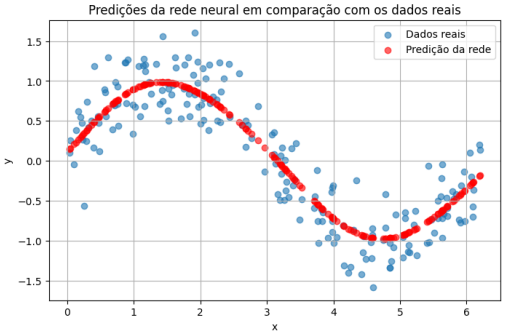
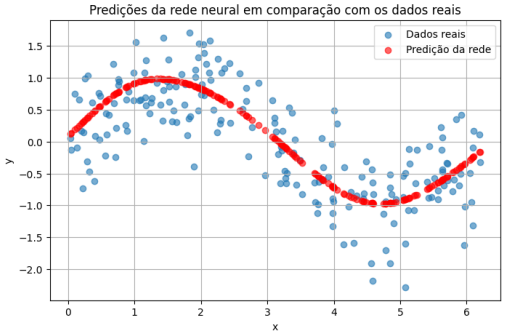


Figura 27 – $\sigma = 0.5$



Em resumo, esta análise destaca a influência do ruído nos dados no processo de treinamento de modelos de aprendizado de máquina. Níveis mais altos de ruído podem levar a uma convergência mais lenta e instável, mas o modelo pode eventualmente adaptar-se e aprender a lidar com o ruído, como observado no caso de $\sigma = 0{,}5$.

2.6 Tarefa 06 – Aplicação de Regularização L2

Comando: Objetivo: Avaliar como a aplicação da regularização L2 (λ_{L2}) influencia o treinamento da rede neural, ajudando a prevenir o *overfitting*. Exploraremos diferentes valores de lambda para entender seu impacto no desempenho da rede.

Foram realizados experimentos de treinamento com quatro diferentes valores de λ_{L2}

(fator de regularização L2): 0,0, 0,001, 0,01 e 0,1. As curvas de treinamento são mostradas na fig. 28. As previsões da rede pode ser visualizadas nas figs. 29 à 32.

Figura 28 – Curvas de aprendizado para diferentes níveis de λ_{L2}

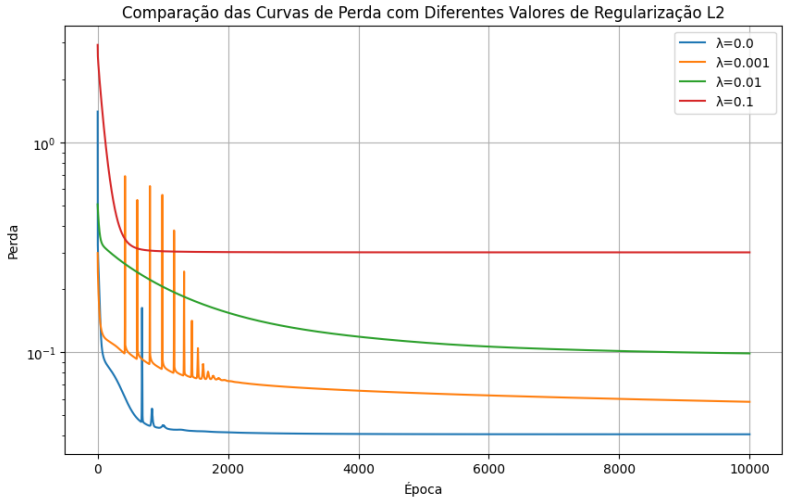


Figura 29 – $\lambda_{L2} = 0,0$

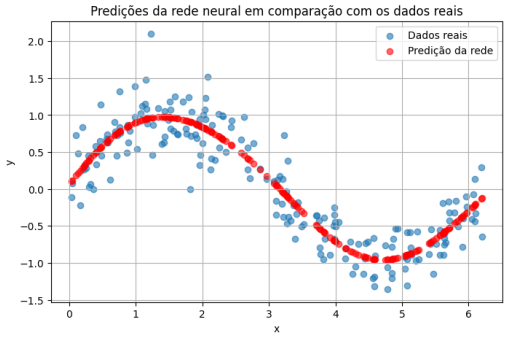


Figura 30 – $\lambda_{L2} = 0,001$

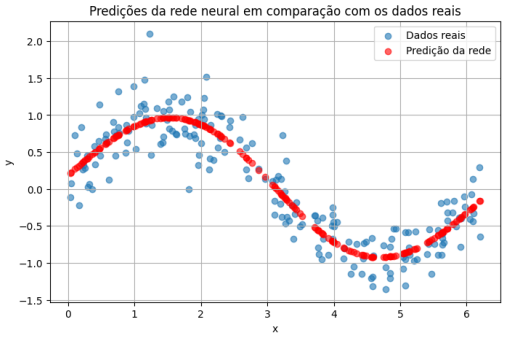


Figura 31 – $\lambda_{L2} = 0,01$

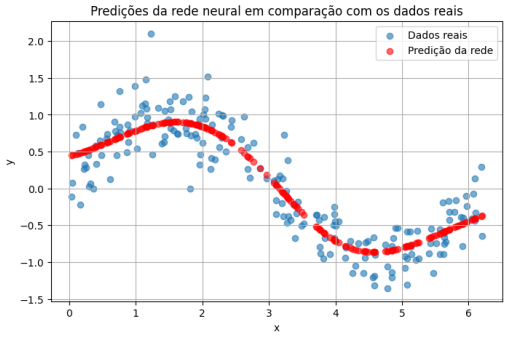
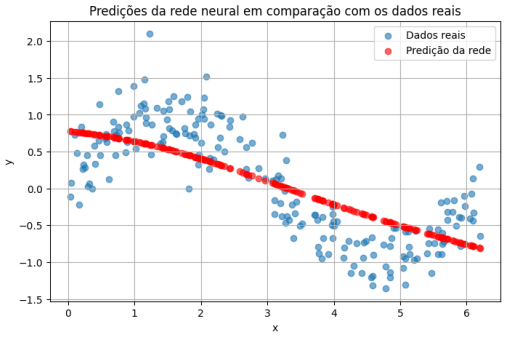


Figura 32 – $\lambda_{L2} = 0,1$



Quando $\lambda_{L2} = 0,0$, o que significa nenhuma regularização, o modelo atinge uma perda de 0.04069 após 9900 épocas. Com $\lambda_{L2} = 0,001$, um pequeno valor de regularização é aplicado, e a perda aumenta para 0.05821. Isso indica que a regularização está tendo um efeito moderado no desempenho do modelo.

Quando $\lambda_{L2} = 0.01$, a perda aumenta significativamente para 0.09901. Neste caso, o modelo está mostrando sinais de *underfitting* (fig. 31), significando que a regularização é

muito forte e começa a prejudicar a capacidade da rede de aprender padrões complexos. O comportamento é semelhante ao de um modelo treinado com poucas épocas (fig. 7), sugerindo que a regularização excessiva pode ter um efeito semelhante à falta de treinamento.

Para $\lambda_{L2} = 0,1$, a perda aumenta drasticamente para 0.30001. Este valor alto de λ_{L2} resulta em uma forte penalidade na complexidade do modelo, levando a um comportamento de previsão quase linear (fig. 32). O modelo está sendo fortemente restrito e não consegue capturar a complexidade dos dados, resultando em um desempenho inferior.

Em resumo, esta análise destaca a importância de escolher o fator de regularização adequado. Um fator muito baixo pode levar a problemas de *overfitting*, enquanto uma regularização muito forte ($\lambda_{L2} = 0.1$) pode resultar em *underfitting* e comportamento simplista. O valor de $\lambda_{L2} = 0,001$ parece ser um bom equilíbrio, permitindo alguma regularização sem comprometer significativamente o desempenho do modelo. A escolha do fator de regularização ideal depende dos dados específicos e do nível de complexidade desejado do modelo.

3 Conclusão

Nesta presente análise, investigamos o impacto de vários parâmetros e hiperparâmetros no treinamento e desempenho de redes neurais. Exploramos diferentes funções de ativação, números de épocas, taxas de aprendizagem, arquiteturas de rede, níveis de ruído nos dados e taxa de regularização. Cada tarefa forneceu "insight"s sobre como esses fatores influenciam o comportamento e a capacidade preditiva das redes neurais.

Esta série de tarefas demonstra a importância da seleção cuidadosa de parâmetros e hiperparâmetros no treinamento de redes neurais. A escolha adequada de funções de ativação, número de épocas, taxa de aprendizagem, arquitetura e regularização é crucial para alcançar um bom desempenho e evitar problemas como *overfitting* e *underfitting*. A análise sistemática desses fatores é essencial para otimizar os modelos de aprendizado de máquina e garantir sua capacidade de generalização para novos dados.