UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
SCC0270 - Redes Neurais e Aprendizado Profundo

Projeto 1
Utilização de MLP

Docente: Prof.ª Dr.ª Roseli Aparecida Romero
Monitor: Diogo Henrique Godoi

Helbert Moreira Pinto - Nº USP 10716504
João Marcos Della Torre Divino - Nº USP 10377708

Abril
2022

Conteúdo

1 Introdução

Este projeto consistiu em estudar o desempenho de uma MLP (*Multi-Layer Perceptron* - Rede Neural Multicamadas, mostrada na Figura 1) em problemas de classificação e regressão, avaliando diferentes composições de hiperparâmetros com o intuíto de encontrar a combinação que fornecesse o melhor resultado em cada caso, ou seja, apresentasse menor erro quadrático médio.

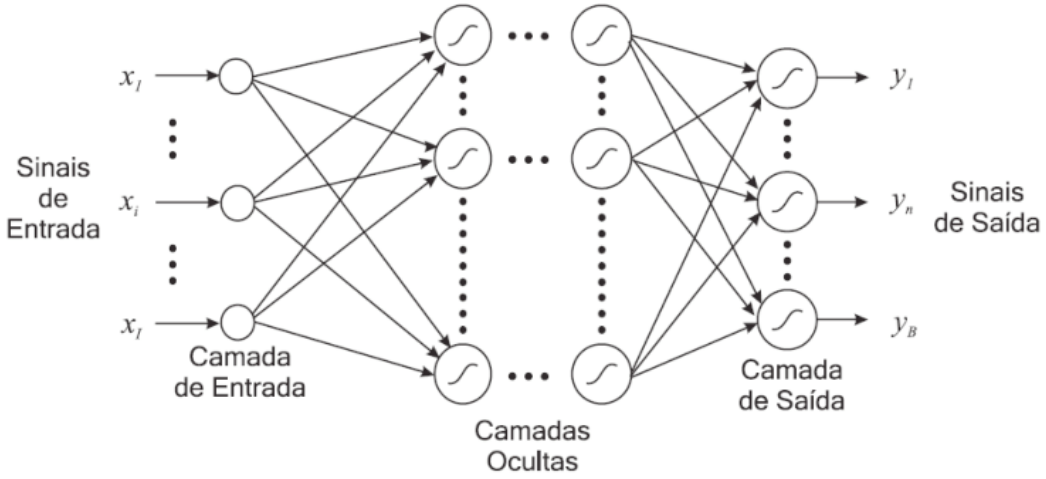

Figura 1: Arquitetura de uma MLP genérica.

Como visto em aula, as MLP's constituem uma ferramenta poderosíssima em diversas áreas da Computação, pois são capazes de solucionarem problemas que exigem soluções não linearmente separáveis, que são os que, em sua maioria, ocorrem na realidade.

Seu diferencial está no algoritmo de aprendizado utilizado, o *backpropagation*, responsável pelo ressurgimento da área de Redes Neurais. Nele, a atualização dos pesos das camadas escondidas leva em conta o gradiente local da camada imediatamente posterior, e não o erro final da rede.

Para este projeto, foi utilizada sua versão generalizada, que proporciona um aumento na velocidade do aprendizado ao inserir um fator α, denominado *momentum*, multiplicado pelo gradiente anterior, no cálculo para atualização dos pesos, como mostra a equação abaixo.

$$\Delta w_{ij}(n) = \eta \delta_j(n) y_i(n) + \alpha \Delta w_{ij}(n-1) \tag{1}$$

Além do ganho de velocidade, consegue-se também reduzir o perigo de instabilidade no aprendizado, podendo evitar que o processo termine em um

mínimo local.

Ora, é de suma importância definir um bom valor para α de maneira a melhorar o desempenho da rede. E não somente ele, mas também outros hiperparâmetros, como a taxa de aprendizado η, a quantidade de camadas e de neurônios em cada uma delas, o conjunto de dados para treino e teste, e por quantas épocas o algoritmo deve rodar, devem ser cuidadosamente calibrados.

Encontrar tais valores, porém, não é trivial, sendo necessário testá-los para cada situação sendo estudada. Nesse sentido, neste projeto, dadas duas bases de dados, uma contendo dados químicos sobre vinhos italianos voltada para classificação, e uma sobre origem de músicas baseado nas características dela, voltada para regressão, buscou-se determinar qual a melhor configuração de rede para cada um.

2 *Datasets* utilizados

A seguir, é feita uma breve apresentação dos *datasets* utilizados neste projeto, ambos obtidos no repositório de *machine learning* da UCI (University of California, Irvine).

2.1 *Wine dataset*

Esta base[1], construída pelo Instituto de Análises e Tecnologias Farmacêuticas e Alimentares de Genoa, na Itália, é composta por dados coletados de uma análise química de 178 amostras de vinhos produzidos numa mesma região, mas de três vinícolas diferentes. Tal análise identificou e determinou 13 constituintes nessas bedidas, sendo eles:

1. Teor alcoólico (%)
2. Ácido málico (g/L)
3. Resíduos de evaporação (mg/L)
4. Alcalinidade dos resíduos (mEq/L)
5. Magnésio (mg/L)
6. Total de fenóis (g/L)
7. Flavonóides (g/L)
8. Fenóis não-flavonóides (g/L)
9. Proantocianins (g/L)
10. Intensidade de cor (valores calculados baseando-se na absorção de luz ultra-violeta)
11. Tonalidade (valores calculados baseando-se na absorção de luz ultra-violeta)
12. OD280 / OD315 de vinhos diluídos - quantidade de proteínas diluídas (g/L)
13. Prolina (mg/L)

[1]Disponível em: https://archive.ics.uci.edu/ml/datasets/Wine

Ora, tais informações correspodem aos atributos de cada instância, sendo eles valores contínuos. Ainda, há um atributo que serve para identificar de qual vinícola procede a amostra, o qual pode assumir o valor 1, 2 ou 3.

É interessante destacar que este *dataset* já foi muito estutado em problemas de classificação; logo, é sabido que as classes são "bem comportadas". Desta forma, espera-se que a MLP não tenha dificuldades para separar os dados.

2.2 *Geographical Origin of Music dataset*

Este *dataset*[2] desenvolvido pelo Professor Fang Zhou na Universidade de Nottingham (Ningbo, China) contém informações a respeito das características do áudio de 1059 músicas provindas de diferentes países do mundo.

Com o intuito de investigar influências regionais nos aspectos da música, foram utilizadas faixas tradicionais mais mundiais, excluindo as populares do Ocidente, cuja influência global já é conhecida.

Ao todo, 68 componentes de cada áudios foram extraídas para montar a base. Além de tais atributos, cada instância possui a informação de latitude e longitude do local de origem.

[2]Disponível em: https://archive.ics.uci.edu/ml/datasets/Geographical+Original+of+Music

3 Desenvolvimento

Com relação aos códigos desenvolvidos para o projeto, estes se encontram organizados da seguinte forma:

- *mlp.py* → Código escrito em python, contendo a implementação base de uma rede MLP;
- *iniciar.ipynb* → Jupyter Notebook contendo a execução principal do projeto.

Já para os conjuntos de dados, a base sobre os vinhos encontra-se no arquivo *wine.data*, enquanto o *dataset* das músicas está localizado no arquivo *default_features_1059_tracks.txt*.

Nas seções seguintes, é descrito como foi realizado o pré-processamento dos dados, como foi feita a implementação e como os casos de estudo foram definidos.

3.1 Pré-processamento dos dados

Antes de alimentar as redes MLP sendo testadas, foi necessário fazer o pré-processamento dos dados. Primeiramente, foi feita a checagem de dados duplicados ou dados faltantes, de maneira a averiguar se algum tratamento era necessário; porém, nenhum desses casos foi identificado.

Em seguida, foi verificado qual o tipo de dado de cada atributo, pois como redes neurais só trabalham com valores numéricos, caracteres ou *strings* teriam de ser transformadas. Contudo, novamente não foi preciso alterá-los, pois todos os dados eram números ou inteiros ou reais.

Após tal checagem inicial, em ambos os *datasets*, foi aplicada a normalização das dados, utilizando o método *MinMax*, para que nenhum atributo tivesse maior peso que os outros. Destaca-se também que, para o *dataset* de vinhos, foi modificado o atributo objetivo (classe), como será explicado na próxima seção.

Feito isso, os dados, para cada base, foram divididos em dois grupos: no primeiro estavam os atributos que serviriam de entrada para a rede; no segundo, aqueles que seriam previstos. E por fim, foi feita a divisão dos dados que seriam utilizados para treinamento e teste das redes.

3.2 Implementação

Com relação à implementação das redes MLP, destacam-se os seguintes pontos:

- Modo padrão para o treinamento da rede: a atualização nos pesos é realizada após a apresentação de cada instância de treinamento;
- Critério de parada: foi definido como o número de épocas, visando verificar de maneira mais direta o impacto do nº de ciclos de treinamento na melhora (ou não) do erro quadratico médio para as amostras de treinamento. Além disso, com as épocas foi possível posteriormente realizar as apresentações com exemplos não vistos pela rede na etapa de treinamento e verificar se os pesos conseguiram identificar os padrões do modelo;
- Pesos inicializados aleatoriamente: foram inicializados os pesos com valores uniformemente distribuídos em um intervalo pequeno (valores com media igual a zero e desvio padrão igual a um $N \sim (0, 1)$);
- Função de ativação: todos os neurônios da rede usam a função sigmoide. Por conta disso, vale mencionar que foi necessário alterar a saída da rede para o caso do *dataset* dos vinhos. Não seria possível ter apenas um neurônio, pois a função de ativação só "abrange"dois valores, e no caso do dataset de vinhos, há temos 3 classes possíveis. Logo, a solução foi colocar 3 neurônios na saída, de maneira que cada um corresponde-se a uma classe. Assim, é possível manter a função sigmoide.
- Realizando alguns testes rápidos, notou-se que, de modo geral, as redes se comportaram melhor quando o número de neurônios presentes nas camadas intermediárias foi reduzido. Desta maneira, optou-se por se realizar testes com $\log(neuronios_{entrada})$ neurônios fixos em todas as camadas intermediárais.

3.3 Casos de estudo

Para conseguir avaliar como a rede se comporta com a presença do termo momentum, realizamos vários testes alterando os hiperparametros e medindo o erro quadratico medio (MSE) para o treino e teste em cada caso.

Para ambos conjuntos de dados, foram utilizados os seguintes hiperparametros:

hiperparametro	valores utilizados
percentual do conjunto utilizado para treino	60% e 80%
taxa de aprendizagem - η	0.1, 0.5 e 0.9
taxa de momentum - α	0, 0.3, 0.5, 0.7 e 1
número de camadas intermediárias	1 e 2
limite máximo de épocas	50 e 200

4 Resultados e discussão

Após a execução de cada caso de teste, coletou-se os resultados e estes foram organizados em tabelas e gráficos para melhor visualização. Eles são apresentados nas seções seguintes, com a respectiva análise de cada *dataset*. Ao final, é feita uma discussão mais geral.

4.1 Análise do *Wine dataset*

Como tivemos muitos casos de teste, montamos uma tabela para melhor organizar os resultados. Abaixo vemos os resultados para o *Wine dataset*.

% teino	80	eta	alpha	MSE		
camadas (intermediarias)	1			Treino	Teste	Δ%
maximo epocas	50	0,1	0,0	0,07551	0,07102	5,9%
			0,3	0,07387	0,07040	4,7%
			0,5	0,07297	0,07012	3,9%
			0,7	0,07217	0,06989	3,2%
			1,0	0,07114	0,06964	2,1%
		0,5	0,0	0,06682	0,07038	-5,3%
			0,3	0,06612	0,07046	-6,6%
			0,5	0,06551	0,07024	-7,2%
			0,7	0,06500	0,07025	-8,1%
			1,0	0,06445	0,07037	-9,2%
		0,7	0,0	0,06487	0,07072	-9,0%
			0,3	0,06379	0,07082	-11,0%
			0,5	0,06296	0,07077	-12,4%
			0,7	0,06100	0,06986	-14,5%
			1,0	0,06097	0,06944	-13,9%

% teino	80	eta	alpha	MSE		
camadas (intermediarias)	2			Treino	Teste	Δ%
maximo epocas	50	0,1	0,0	0,08522	0,08090	5,1%
			0,3	0,08311	0,07835	5,7%
			0,5	0,08185	0,07701	5,9%
			0,7	0,08073	0,07589	6,0%
			1,0	0,07905	0,07436	5,9%
		0,5	0,0	0,06945	0,07059	-1,6%
			0,3	0,06707	0,07075	-5,5%
			0,5	0,06590	0,07091	-7,6%
			0,7	0,06504	0,07116	-9,4%
			1,0	0,06416	0,07186	-12,0%
		0,7	0,0	0,06522	0,07180	-10,1%
			0,3	0,06391	0,07270	-13,8%
			0,5	0,06342	0,07348	-15,9%
			0,7	0,06320	0,07423	-17,5%
			1,0	0,06326	0,07535	-19,1%

Figura 2: Resultados classificação Wine para 50 epocas

% teino	80	eta	alpha	MSE		
camadas (intermediarias)	1			Treino	Teste	Δ%
maximo epocas	200	0,1	0,0	0,06466	0,07002	-8,3%
			0,3	0,06251	0,07013	-12,2%
			0,5	0,06132	0,07026	-14,6%
			0,7	0,06039	0,07051	-16,8%
			1,0	0,05937	0,07094	-19,5%
		0,5	0,0	0,05464	0,07599	-39,1%
			0,3	0,05120	0,07334	-43,2%
			0,5	0,05093	0,07318	-43,7%
			0,7	0,05103	0,07340	-43,8%
			1,0	0,05076	0,07457	-46,9%
		0,7	0,0	0,05082	0,07305	-43,7%
			0,3	0,05024	0,07325	-45,8%
			0,5	0,05024	0,07530	-49,9%
			0,7	0,05051	0,07436	-47,2%
			1,0	0,05070	0,07455	-47,0%

% teino	80	eta	alpha	MSE		
camadas (intermediarias)	1			Treino	Teste	Δ%
maximo epocas	200	0,1	0,0	0,06880	0,07005	-1,8%
			0,3	0,06577	0,07022	-6,8%
			0,5	0,06420	0,07040	-9,7%
			0,7	0,06293	0,07053	-12,1%
			1,0	0,06145	0,07071	-15,1%
		0,5	0,0	0,05487	0,07328	-33,6%
			0,3	0,05319	0,07298	-37,2%
			0,5	0,05271	0,07396	-40,3%
			0,7	0,05240	0,07521	-43,5%
			1,0	0,05197	0,07779	-49,7%
		0,7	0,0	0,05256	0,07473	-42,2%
			0,3	0,05199	0,07878	-51,5%
			0,5	0,05165	0,08039	-55,6%
			0,7	0,05161	0,08308	-61,0%
			1,0	0,05274	0,08912	-69,0%

Figura 3: Resultados classificação Wine para 200 epocas

Inicialmente comparando os resultados em cada uma das tabelas, percebemos que há um padrão em relação à diferença percentual do erro quadratico médio do teste em relação ao do treino ΔMSE. Vemos que ao aumentar o

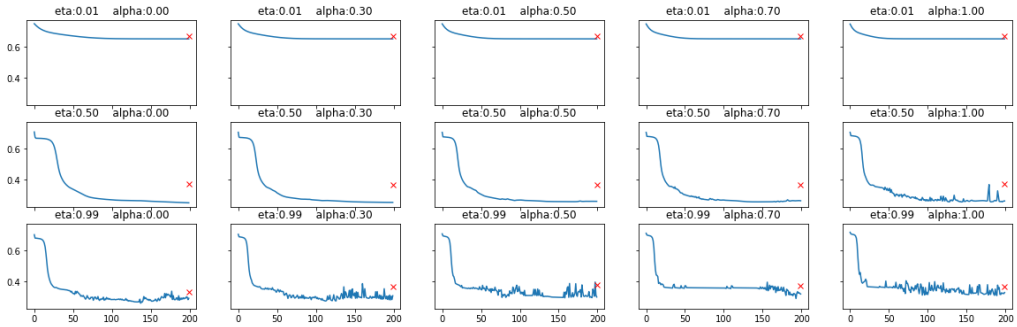

Figura 4: MSE da classificação Wine com 2 camadas intermediárias com 1 neurônio cada

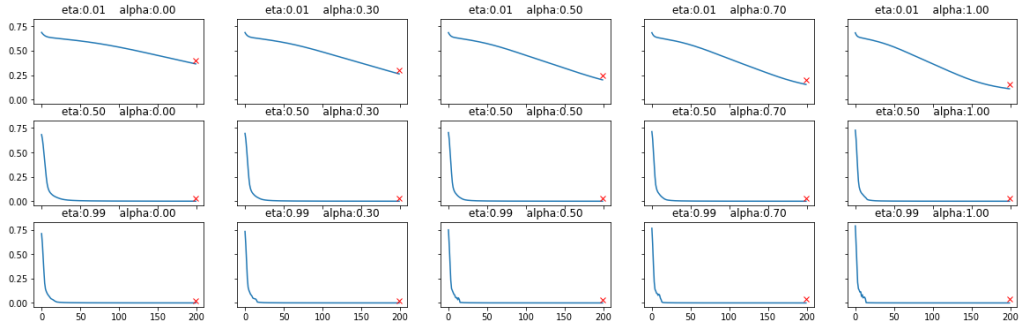

Figura 5: MSE da classificação Wine com 2 camadas intermediárias com 10 neurônios cada

valor de η, podemos até reduzir o erro absoluto, porem aumentados a diferença do MSE entre treino/teste. Isso é explicado pelo fato de que quando aumentamos o valor de η estamos dando uma importancia maior para o conjunto de treino, melhorando o valor absoluto do MSE. Entretanto quando medimos o erro no conjunto de testes, vemos que há uma melhora, numa taxa menor, resultando no aumento da diferença percentual.

O valor de α colabora para o mesmo fenomeno, so que como α adiciona uma parcela de ∇W_{t-1}, se ∇W_{t-1} for pequeno o impacto é menor. Em outras palavras, o impacto do α é menor se η é baixo.

Quando analisamos os casos, comparando uma e duas camadas intermediárias, vemos que para α baixo, os valores do MSE são, quando utilizamos apenas uma camada, em média maiores que o dobro se comparados ao teste com duas camadas. Só que ao aumentar o valor de α vemos que o MSE em ambos casos (com uma e duas camadas) vão ficando mais próximos, ao ponto de que não podemos notar uma diferença significativa.

Se compararmos os resultados com máximo de épocas de 50 para 200, vemos que o MSE cai em média uma ordem de grandeza no conjunto de treino e numa taxa menor para os casos de teste. Em outras palavras, para um aumento de 4x no processamento de teste, tivemos uma redução de aproximadamente 10% do MSE (de treino).

4.2 Análise do *Geographical Origin of Music dataset*

Como tivemos muitos casos de teste, montamos uma tabela para melhor organizar os resultados. Abaixo vemos os resultados para o *Geographical Origin of Music dataset*.

% teino	80	eta	alpha	MSE		
camadas (intermediarias)	1			Treino	Teste	Δ%
maximo epocas	50	0,1	0,0	0,10986	0,12962	-18,0%
			0,3	0,07547	0,10253	-35,9%
			0,5	0,06227	0,09170	-47,3%
			0,7	0,05291	0,08358	-58,0%
			1,0	0,04299	0,07427	-72,8%
		0,5	0,0	0,01469	0,04153	-182,7%
			0,3	0,01136	0,03701	-225,8%
			0,5	0,00991	0,03530	-256,2%
			0,7	0,00880	0,03423	-289,0%
			1,0	0,00750	0,03343	-345,7%
		0,7	0,0	0,00837	0,03786	-352,3%
			0,3	0,00640	0,03785	-491,4%
			0,5	0,00559	0,03936	-604,1%
			0,7	0,00437	0,03690	-744,4%
			1,0	0,00285	0,04134	-1350,5%

% teino	80	eta	alpha	MSE		
camadas (intermediarias)	2			Treino	Teste	Δ%
maximo epocas	50	0,1	0,0	0,33730	0,39507	-17,1%
			0,3	0,22807	0,26771	-17,4%
			0,5	0,13272	0,14181	-6,8%
			0,7	0,08407	0,08053	4,2%
			1,0	0,05146	0,04731	8,1%
		0,5	0,0	0,00716	0,01375	-92,0%
			0,3	0,00506	0,01525	-201,4%
			0,5	0,00411	0,02137	-420,0%
			0,7	0,00353	0,02901	-721,8%
			1,0	0,00711	0,04169	-486,4%
		0,7	0,0	0,00665	0,03686	-454,3%
			0,3	0,00645	0,04764	-638,6%
			0,5	0,00218	0,02706	-1141,3%
			0,7	0,00061	0,01355	-2121,3%
			1,0	0,00065	0,02355	-3523,1%

Figura 6: Resultados regressão Geographical Origin of Music para 50 epocas

% teino	80	eta	alpha	MSE		
camadas (intermediarias)	1			Treino	Teste	Δ%
maximo epocas	200	0,1	0,0	0,01617	0,05285	-226,8%
			0,3	0,01150	0,05000	-334,8%
			0,5	0,00951	0,04927	-418,1%
			0,7	0,00803	0,04899	-510,1%
			1,0	0,00642	0,04904	-663,9%
		0,5	0,0	0,00175	0,05393	-2981,7%
			0,3	0,00120	0,05567	-4539,2%
			0,5	0,00098	0,05657	-5672,4%
			0,7	0,00081	0,05732	-6976,5%
			1,0	0,00064	0,05830	-9009,4%
		0,7	0,0	0,00075	0,05704	-7505,3%
			0,3	0,00051	0,05855	-11380,4%
			0,5	0,00041	0,05998	-14529,3%
			0,7	0,00034	0,05984	-17500,0%
			1,0	0,00028	0,06258	-22250,0%

% teino	80	eta	alpha	MSE		
camadas (intermediarias)	2			Treino	Teste	Δ%
maximo epocas	200	0,1	0,0	0,00841	0,03017	-258,7%
			0,3	0,00494	0,03477	-603,8%
			0,5	0,00378	0,03770	-897,4%
			0,7	0,00302	0,04025	-1232,8%
			1,0	0,00228	0,04343	-1804,8%
		0,5	0,0	0,00056	0,05711	-10098,2%
			0,3	0,00038	0,05633	-14723,7%
			0,5	0,00031	0,05677	-18212,9%
			0,7	0,00026	0,05854	-22415,4%
			1,0	0,00020	0,05349	-26645,0%
		0,7	0,0	0,00024	0,05430	-22525,0%
			0,3	0,00016	0,05779	-36018,8%
			0,5	0,00014	0,01721	-12192,9%
			0,7	0,00013	0,02098	-16038,5%
			1,0	0,00011	0,01707	-15418,2%

Figura 7: Resultados regressão Geographical Origin of Music para 200 epocas

Diferente dos casos do tópico anterior, vemos que o α pouco impacta no ΔMSE do teste em relação ao do treino. Entretando o valor de η é muito mais representativo neste quesito. Inclusive para valores baixos de η, tivemos casos onde os dados de teste obtiveram um MSE inferior ao registrado no

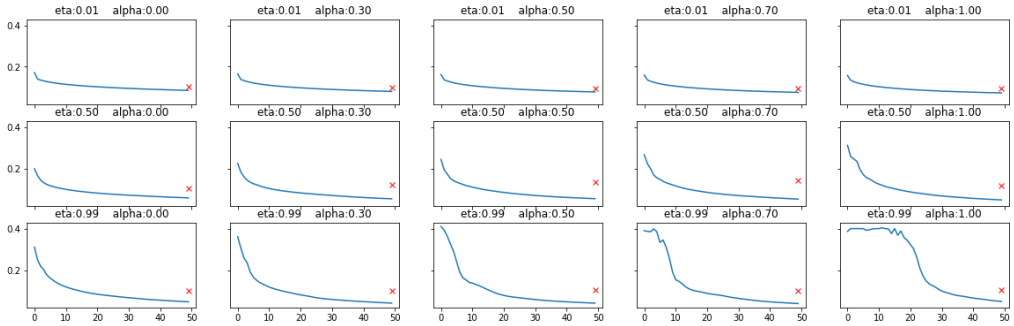

Figura 8: MSE da regressão Geographical Origin of Music com 1 camada intermediária com 100 neurônios

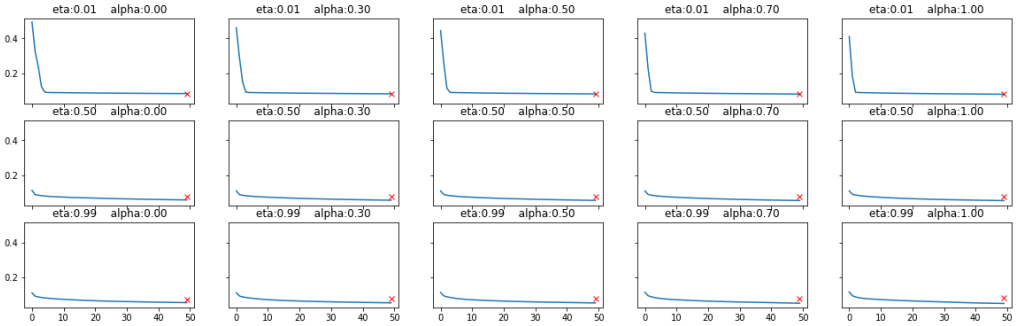

Figura 9: MSE da regressão Geographical Origin of Music com 2 camadas intermediárias com 10 neurônios cada

treino. Isso se deve ao fato de se tratar de uma regressão, onde o $y_{esperado}$ é um valor real e ter uma baixa taxa de aprendizage, neste caso, significa aproximar mais lentamente em relação ao ótimo global.

Como esperado vemos em ambos os datasets que o MSE é maior nos casos onde utilizamos apenas 60% dos dados para treino em relação aos casos onde utilizamos 80%. Por fim, vemos que o ΔMSE mantem-se estavel ao aumentar o valor de α quando utilizamos redes com duas camadas intermediárias.

4.3 Análise geral dos resultados

Como executamos os projetos com vários valores para os hiperparametros, percebemos que o desempenho geral da rede com duas camadas pouco melhorou em relação aos resultados com apenas uma camada (o problema de regressão teve um resultado pouco melhor, mas nada muito surpreendente), de modo que não vale muito a pena aumentar o tempo de processamento para baixar alguns decimos do MSE.

Outro ponto que pode-se observar a partir dos resultados é que utilizando um η muito próximo a zero, a curva de minimização da função de custo torna-se quase uma finção constante. Neste caso, o tempo para se obter um valor limite de erro é bem maior do que quando utilizamos um η maior. O problema de regressão foi menos impactado com valores pequenos de η. Entretanto quando utilizamos valores altos para o η (muito proximos a um), vemos que as redes que possuem muitos neurônios nas camadas intermediárias tornaram-se instáveis, mantendo o MSE no mesmo valor por todo o treinamento. Com relação ao α vimos que sua função é auxiliar na convergencia da rede, fazendo com que a curva do MSE x epocas ficasse mais verticalizada, principalmente nas primeiras epocas. No entanto, quando testamos a rede com o valor um para α, praticamente todos os casos de teste ficaram instáveis.

Outro ponto que devemos considerar é a quantidade de neuronios nas camadas intermediarias, onde vemos nos resultados que para ambos os problemas, os resultados são significativamente piores quando inserimos muitos neuronios nas camadas intermediarias (ou pelo menos, quando adicionamos mais neurônios que o necessário). O problema de regressão funcionou bem com 10 neuronios na camada intermediaria, entretanto performou melhor quando inserimos apenas um neuronio nas camadas intermediarias. O problema de classificação teve um melhor desempenho utilizando 10 neuronios na camada intermediaria.

Por fim, vemos no grafico de resultados que em ambos os problemas (classificação e regressão) houve valores utilizados ruins para η e/ou α fazendo a rede ficar por varias epocas sem conseguir reduzir o valor da função de custo. Interpretamos este fenômeno considerando que a rede encontrou minimos locais, mantendo o valor do erro sem redução.

5 Conclusão

Uma questão que levantamos foi a seguinte: "Se no caso da classificação utilizamos três neurônios para saída, e no de regressão estamos utilizamos apenas um, porque o valor absoluto do MSE na regressão é maior, visto que a função de custo soma os erros dos neurônios da saída?". Não temos uma resposta direta para este questionamento, porém temos duas hiposteses.

A primeira é de que os dados utilizados na classificação são de fato mais bem separados entre os grupos, então a rede MLP conseguiu com o mesmo número de epocas um MSE menor.

A segunda hipotese é de que a função de ativação aproxima melhor quando o y_{real} está em um dos extremos, pois assim o ajuste nos pesos não precisa ser necessariamente refinado para se obter saídas nos extremos e portanto próximo à saída.

Por fim, os valores ideais para η e α não são fixos como os utilizados no desenvolvimento do trabalho e sim variáveis com o decorrer das épocas. Em épocas iniciais, espera-se que tenhamos um erro maior, pois as primeiras saídas da rede MLP são baseadas em pesos randômicos. Neste caso um η alto auxilia a minimizar mais rapidamente a função de custo. Entretanto um η alto em épocas avançadas dificultaria a aproximação do ponto de mínimo global, pois o η faz com que o valor de ∇W seja alto.

No caso do α temos a situação que em épocas iniciais a parcela de momentum ∇W_{t-1} pode ou não ter o mesmo sinal de ∇W (depende de como foram iniciados os pesos randômincos). Se os valores tiverem mesmo sinal até podemos considerar que isso poderia auxiliar a reduzir mais rapidamente a função de custo, só que o valor do momentum depende de ∇W que possui valor associado a η. Agora se os valores tiverem sinal invertido, um α alto fará com que a função de custo seja minimizada a uma taxa menor em direção ao ótimo, atrapalhando o tempo de processamento da rede. Já em estágios avançados temos o caso em que estamos muito proximos do ponto que minimiza a função de custo e neste ponto se ∇W é grande, um α grande ajuda a minimizar o passo para que a função se aproxime de modo mais suave do ponto que minimiza a função de custo.

Resumindo, o caso ideal seria um α que vai crecendo e um η que decresce ao longo das epocas.