

Nome: _____

Quadro de respostas (questões objetivas):

Questão	a	b	c	d	e	f	g	h
1								
2								
3								
4								
5								

Questões Objetivas (6.0 pts)

Cada questão possui apenas uma alternativa correta. Marque a letra escolhida no quadro de respostas.

- 1) (1.2) Por que uma rede de Perceptrons de Rosenblatt, na sua formulação original com saída $\hat{y} = \text{sign}(\mathbf{x}^\top \boldsymbol{\theta})$, não pode ser organizada em uma rede multicamadas treinada por descida de gradiente e *backpropagation*?
- a) Porque o produto escalar $\mathbf{x}^\top \boldsymbol{\theta}$ não é diferenciável em relação aos pesos $\boldsymbol{\theta}$.
 - b) Porque a derivada de $\text{sign}(z)$ é zero em $z \neq 0$ e indefinida em $z = 0$, bloqueando a regra da cadeia.
 - c) Porque o Perceptron tem como limitação modelar apenas problemas linearmente separáveis; com função de ativação contínua, esse problema desapareceria automaticamente.
 - d) Porque a função sign é descontínua, o que impede que a função de custo de entropia cruzada seja calculada.
 - e) Porque a regra de atualização original do PLA (*Perceptron Learning Algorithm*) é incompatível com descida de gradiente; basta substituir o PLA pelo SGD para o problema desaparecer.
 - f) Porque o número de parâmetros do Perceptron é grande demais para ser otimizado por descida de gradiente em hardware atual.
 - g) Porque o Perceptron já realiza descida de gradiente internamente; encadeá-los em rede multicamadas dobraria a operação e levaria a divergência.
 - h) Porque a saída $\hat{y} \in \{-1, +1\}$ não é interpretável como probabilidade, e a descida de gradiente exige saídas probabilísticas.

2) (1.2) Sobre as asserções abaixo a respeito de funções de ativação:

I. A função sigmoide $\sigma(z) = 1/(1+e^{-z})$ satura para entradas com módulo grande, fazendo com que a sua derivada tenda a zero. Em redes profundas, isso contribui para o problema do desaparecimento do gradiente nas camadas iniciais.

II. A função ReLU é estritamente positiva para qualquer entrada, garantindo que nenhuma unidade da rede possa “morrer” (deixar permanentemente de produzir saída diferente de zero) durante o treinamento.

III. A derivada da função tanh tem magnitude máxima maior do que a da sigmoide ($\tanh'(0) = 1$ contra $\sigma'(0) = 0.25$), de modo que, em regiões não saturadas, o gradiente propagado é tipicamente maior, o que costuma permitir uma convergência mais rápida do treinamento em comparação com a sigmoide.

- a) Apenas I está correta.
- b) Apenas II está correta.
- c) Apenas III está correta.
- d) Apenas I e II estão corretas.
- e) Apenas I e III estão corretas.
- f) Apenas II e III estão corretas.
- g) Todas estão corretas.
- h) Nenhuma está correta.

3) (1.2) Você está treinando um MLP profundo com 10 camadas ocultas para um problema de classificação binária. Optou por sigmoide em todas as camadas ocultas, sigmoide na camada de saída, função de custo entropia cruzada binária, otimizador SGD com taxa de aprendizado padrão e inicialização $\mathcal{N}(0, 1)$ para todos os pesos. Após muitas épocas, o erro de treino permanece alto e os gradientes em relação aos pesos das primeiras camadas são extremamente próximos de zero. Qual a explicação mais consistente e a melhor mitigação?

- a) A taxa de aprendizado está baixa demais; aumentá-la em duas ordens de grandeza compensa os gradientes pequenos e resolve o problema.
- b) O modelo está sofrendo *overfitting*; reduzir o número de camadas para uma única camada oculta resolve o problema.
- c) Os gradientes nas camadas iniciais ficam próximos de zero devido à saturação da sigmoide nas camadas profundas; trocar a sigmoide das ocultas por ReLU com inicialização de He resolve.
- d) A função de custo entropia cruzada binária é inadequada para classificação; substituí-la por L2 loss resolve o problema.
- e) Os pesos devem ser inicializados em zero para evitar que comecem em regiões saturadas da sigmoide.
- f) Trocar a sigmoide da camada de saída por SoftMax resolve o desaparecimento do gradiente nas camadas iniciais.
- g) Adicionar Dropout com probabilidade alta nas camadas iniciais aumenta o gradiente nessas camadas.
- h) O problema decorre exclusivamente do otimizador SGD; substituí-lo por Adam, sem outras alterações, é suficiente para que o erro caia.

- 4) (1.2) Por que, na prática, a função de ativação da camada de saída de uma rede neural é escolhida *em conjunto* com a função de custo (por exemplo, sigmoide com entropia cruzada binária; SoftMax com entropia cruzada multinomial)?
- a) Porque é a única forma de evitar *overfitting* na camada de saída.
 - b) Porque sem essa combinação a rede é incapaz, em princípio, de produzir saídas no intervalo correto (por exemplo, $[0, 1]$ para sigmoide).
 - c) Porque o algoritmo de *backpropagation* só funciona quando função de custo e ativação da saída são compostas algebricamente.
 - d) Porque o teorema da aproximação universal exige especificamente esse pareamento para garantir convergência da descida de gradiente.
 - e) Porque essa combinação é o único caso em que a função de custo é convexa em relação aos pesos da rede como um todo.
 - f) Porque essa combinação simplifica o gradiente em relação aos *logits*, resultando em $\partial \mathcal{L} / \partial z = \hat{y} - y$.
 - g) Porque essa combinação reduz o número de parâmetros treináveis na última camada.
 - h) Porque, sem ela, a derivada da função de custo em relação aos pesos da última camada é sempre zero.
- 5) (1.2) Sobre a inicialização dos pesos de uma rede neural profunda com função de ativação ReLU, qual afirmação é correta?
- a) Inicializar todos os pesos com zero é adequado para ReLU; o gradiente assimétrico da ReLU faz com que diferentes unidades aprendam características distintas mesmo partindo de pesos idênticos.
 - b) Inicializar todos os pesos com a mesma constante não-nula é adequado, desde que a constante seja calibrada em função do número de camadas.
 - c) A inicialização de Xavier (variância $1/n_{\text{in}}$ ou $2/(n_{\text{in}} + n_{\text{out}})$) é a recomendada para ReLU, pois assume que a função de ativação se comporta de forma aproximadamente linear.
 - d) A inicialização de He (variância $2/n_{\text{in}}$) é recomendada para ReLU, compensando o fato de cerca de metade das pré-ativações serem zeradas.
 - e) Qualquer inicialização aleatória uniforme em $[-1, 1]$ funciona igualmente bem, independentemente da profundidade da rede e da função de ativação utilizada.
 - f) Para ReLU, recomenda-se uma inicialização determinística baseada na projeção PCA do conjunto de treinamento sobre os pesos da primeira camada.
 - g) A inicialização aleatória dos pesos é desnecessária; basta inicializar os vieses (*biases*) com valores aleatórios para quebrar a simetria.
 - h) Pesos amostrados de $\mathcal{N}(0, 1)$ (variância 1, sem reescala) são apropriados para qualquer arquitetura com ReLU, desde que a taxa de aprendizado seja suficientemente pequena.
-

Questões Dissertativas (4.0 pts)

6) (2.0) Marque **Verdadeiro** ou **Falso** nas asserções abaixo. Em caso **Falso**, proponha uma modificação *não-trivial* que torna a asserção verdadeira (a simples negação **não** é uma modificação válida).

- a) ☐ V ☐ F Um MLP com 1 unidade de entrada, K camadas ocultas com D unidades cada (todas com viés) e 1 unidade de saída possui exatamente $3D+1+(K-1)D^2$ parâmetros treináveis.

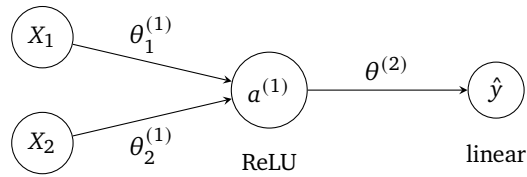
- b) ☐ V ☐ F A descida de gradiente em lote (*batch gradient descent*) calcula o gradiente sobre uma única instância (ou um *mini-batch* pequeno) a cada iteração; já a SGD (*Stochastic Gradient Descent*) calcula o gradiente da função de custo sobre todo o conjunto de treinamento a cada iteração.

- c) ☐ V ☐ F O problema do desaparecimento do gradiente é particularmente severo com a função de ativação sigmoide em redes profundas, pois a derivada $\sigma'(z) = \sigma(z)(1 - \sigma(z))$ tem valor máximo 0.25 e tende a zero quando a sigmoide satura (i.e., para $|z|$ grande), fazendo com que o gradiente decaia multiplicativamente ao longo das camadas.

- d) ☐ V ☐ F A função de custo de entropia cruzada binária pode ser obtida a partir da estimação por máxima verossimilhança dos rótulos $y_i \in \{0, 1\}$ assumindo que cada rótulo segue uma distribuição *Gaussiana* de média $\hat{y}_i$ e variância fixa.

- e) ☐ V ☐ F Inicializar todos os pesos de uma camada oculta com zero é adequado, pois o algoritmo de *backpropagation* naturalmente quebra a simetria entre as unidades nas primeiras iterações da descida de gradiente.

- 7) (2.0) Considere o MLP abaixo, com função de ativação ReLU na camada oculta (1 unidade) e saída linear. As entradas são $X_1 = 1$ e $X_2 = 2$, e a saída esperada é $y = 5$. Os pesos iniciais são $\theta_1^{(1)} = 1$, $\theta_2^{(1)} = 1$ e $\theta^{(2)} = 2$ (todos sem viés). A função de custo é $J(\theta) = \frac{1}{2}(y - \hat{y})^2$.



- (0.5) Compute o *forward pass* e indique o valor de $\hat{y}$.
- (0.5) Compute o valor da função de custo $J(\theta)$.
- (0.5) Compute os gradientes $\frac{\partial J}{\partial \theta^{(2)}}$, $\frac{\partial J}{\partial \theta_1^{(1)}}$ e $\frac{\partial J}{\partial \theta_2^{(1)}}$.
- (0.5) Realize uma atualização dos três parâmetros usando descida de gradiente estocástica $\theta_{t+1} = \theta_t - \eta \nabla_{\theta} J$ com taxa de aprendizado $\eta = 0.1$.