Redes Neurais e Backpropagation

12 de julho de 2023

1 Regra da cadeia

Claro, vou explicar a regra da cadeia utilizando a notação LaTeX. A regra da cadeia é um teorema fundamental no cálculo diferencial que é usado quando precisamos diferenciar a composição de duas ou mais funções.

Seja $y = f(g(x))$ uma composição de duas funções $f(u)$ e $g(x)$, aonde $u = g(x)$. A regra da cadeia estabelece que a derivada de y em relação a x é a derivada de f em relação a u multiplicada pela derivada de g em relação a x. Em notação matemática, a regra da cadeia é expressa da seguinte maneira:

$$\frac{dy}{dx} = \frac{df}{du} \cdot \frac{du}{dx}$$

Para extender isso para funções de múltiplas variáveis, digamos que temos $y = f(g(x), h(x))$, a regra da cadeia para funções de múltiplas variáveis é:

$$\frac{dy}{dx} = \frac{\partial f}{\partial g} \cdot \frac{dg}{dx} + \frac{\partial f}{\partial h} \cdot \frac{dh}{dx}$$

Aqui, $\frac{\partial f}{\partial g}$ e $\frac{\partial f}{\partial h}$ representam as derivadas parciais de f com respeito a g e h respectivamente.

2 Backpropagation para uma rede simples

Vamos considerar uma rede neural com 3 camadas e um neurônio em cada camada. Vamos supor que as funções de ativação em cada neurônio são funções sigmoid. A função de custo será o erro quadrático médio.

A ativação em cada neurônio é denotada por a e os pesos e bias por w e b respectivamente. As funções de ativação da camada i serão denotadas por a_i, e o mesmo vale para os pesos e os bias w_i e b_i. O sinal de entrada para um neurônio é denotado por z, então $z_i = w_i a_{i-1} + b_i$ e $a_i = \sigma(z_i)$, onde σ é a função de ativação sigmoid.

Para o caso do erro quadrático médio, a função de custo C é dada por $C = \frac{1}{2}(y - a_3)^2$, onde y é o valor desejado de saída e a_3 é a saída da rede. Usando a regra da cadeia, podemos calcular o gradiente do erro em relação aos pesos e bias. Vamos começar com a última camada:

$$\frac{\partial C}{\partial w_3} = \frac{\partial C}{\partial a_3} \cdot \frac{\partial a_3}{\partial z_3} \cdot \frac{\partial z_3}{\partial w_3}$$

Calculando cada parte individualmente:

1. $\frac{\partial C}{\partial a_3} = a_3 - y$
2. $\frac{\partial a_3}{\partial z_3} = \sigma'(z_3) = \sigma(z_3)(1 - \sigma(z_3)) = a_3(1 - a_3)$
3. $\frac{\partial z_3}{\partial w_3} = a_2$

Combinando todas essas partes, nós obtemos:

$$\frac{\partial C}{\partial w_3} = (a_3 - y) \cdot a_3(1 - a_3) \cdot a_2$$

De forma similar, podemos calcular a derivada parcial em relação ao bias b_3:

$$\frac{\partial C}{\partial b_3} = (a_3 - y) \cdot a_3(1 - a_3)$$

O mesmo processo pode ser aplicado às camadas anteriores, sempre lembrando que a derivada do custo em relação à ativação da camada anterior agora também depende das derivadas das camadas seguintes. Para a camada 2, por exemplo, temos:

$$\frac{\partial C}{\partial w_2} = \frac{\partial C}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_2} \cdot \frac{\partial z_2}{\partial w_2}$$

Onde agora:
1. $\frac{\partial C}{\partial a_2} = \frac{\partial C}{\partial a_3} \cdot \frac{\partial a_3}{\partial z_3} \cdot \frac{\partial z_3}{\partial a_2} = (a_3 - y) \cdot a_3(1 - a_3) \cdot w_3$ 2. $\frac{\partial a_2}{\partial z_2} = \sigma'(z_2) = a_2(1 - a_2)$ 3. $\frac{\partial z_2}{\partial w_2} = a_1$

Então,

$$\frac{\partial C}{\partial w_2} = (a_3 - y) \cdot a_3(1 - a_3) \cdot w_3 \cdot a_2(1 - a_2) \cdot a_1$$

E

$$\frac{\partial C}{\partial b_2} = (a_3 - y) \cdot a_3(1 - a_3) \cdot w_3 \cdot a_2(1 - a_2)$$

O processo pode ser aplicado à camada 1 de maneira análoga, levando em consideração as derivadas das camadas seguintes.

3 Exercícios

1. **Funções de Ativação**: Descreva o propósito de uma função de ativação em uma rede neural e discuta as diferenças entre a função sigmóide, ReLU (Rectified Linear Unit) e a função tangente hiperbólica. Qual seria apropriado usar em diferentes contextos?

2. **Rede Neural para Regressão Linear Simples**: Suponha que você deseja treinar uma rede neural para realizar regressão linear simples (ou seja, há apenas uma variável de entrada e uma variável de saída). Desenhe a estrutura da rede (quantos neurônios, camadas, que tipo de função de ativação, etc.). Qual seria a função de custo adequada neste caso?

3. **Backpropagation**: Suponha que você tenha uma rede neural feed-forward com 2 camadas ocultas. Cada camada oculta tem 2 neurônios e a camada de saída tem um único neurônio. A função de ativação é a função sigmóide e a função de custo é o erro quadrático médio. O vetor de entrada é (1,0), o vetor de pesos da primeira camada para a segunda é ((0.5, 0.3),(0.1,0.2)), o vetor de bias da primeira camada para a segunda é (0.3,0.2), o vetor de pesos da segunda camada para a saída é (0.4,0.6) e o bias da segunda camada para a saída é 0.1. A saída desejada é 1. Calcule a saída da rede e em seguida, calcule a atualização dos pesos e bias por meio de backpropagation com uma taxa de aprendizado de 0.1.

4 Soluções

1. **Funções de Ativação**: As funções de ativação são usadas em redes neurais para introduzir não linearidades. Eles ajudam a rede a aprender a partir de dados complexos e a fazer previsões precisas.

   - **Sigmóide**: A função sigmóide tem uma curva em forma de "S", variando entre 0 e 1. Ela é útil para modelos de classificação binária onde queremos que a saída seja uma probabilidade. No entanto, a função sigmóide sofre do problema de "desaparecimento do gradiente", o que significa que para valores muito grandes ou muito pequenos, o gradiente é quase zero e a rede aprende muito lentamente.

   - **ReLU (Rectified Linear Unit)**: A função ReLU retorna 0 para valores negativos e o valor de entrada para valores positivos. Ela ajuda a resolver o problema do desaparecimento do gradiente e é computacionalmente eficiente, sendo por isso a função de ativação mais comumente usada em redes neurais convolucionais e profundas. No entanto, pode sofrer do problema de "neurônios mortos", onde certos neurônios nunca são ativados.

   - **Tangente Hiperbólica (tanh)**: A função tanh é semelhante à sigmóide, mas varia de -1 a 1. Ela é útil quando queremos que as saídas negativas sejam consideradas. No entanto, como a sigmóide, também sofre do problema do desaparecimento do gradiente.

2. **Rede Neural para Regressão Linear Simples**: Para realizar uma regressão linear simples, você precisaria de uma rede neural com apenas uma camada de entrada, uma camada de saída e nenhuma camada oculta. A camada de entrada teria um único neurônio (para a única variável de entrada), e a camada de saída também teria um único neurônio (para a variável de saída). Não haveria necessidade de uma função de ativação, uma vez que a saída é uma função linear da entrada. A função de custo apropriada seria o erro quadrático médio, que é comumente usado para problemas de regressão.

3. **Backpropagation**:

   A saída da rede é calculada da seguinte forma:

   - Para a primeira camada oculta: $z_1 = w_1 \cdot x + b_1 = (0.5 \cdot 1 + 0.3 \cdot 0) + 0.3 = 0.8$ e $z_2 = w_2 \cdot x + b_2 = (0.1 \cdot 1 + 0.2 \cdot 0) + 0.2 = 0.3$. - Aplicamos a função de ativação sigmóide para obter as ativações: $a_1 = \sigma(z_1) = \frac{1}{1+e^{-0.8}} \approx 0.69$ e $a_2 = \sigma(z_2) = \frac{1}{1+e^{-0.3}} \approx 0.57$. - Para a segunda camada oculta: $z_3 = w_3 \cdot a + b_3 = (0.4 \cdot a_1 + 0.6 \cdot a_2) + 0.1 = 0.55$. - Aplicamos a função de ativação sigmóide para obter a ativação: $a_3 = \sigma(z_3) \approx 0.63$.

   A atualização dos pesos e bias através da retropropagação é calculada da seguinte forma:

   - O erro é $e = \frac{1}{2}(y - a_3)^2 = \frac{1}{2}(1 - 0.63)^2 = 0.0684$. - O gradiente é $\delta_3 = (a_3 - y) \cdot a_3(1 - a_3) = -0.0874$. - As atualizações de peso são $\Delta w_3 = -\eta \cdot a_2 \cdot \delta_3 = 0.0048$ e $\Delta w_2 = -\eta \cdot a_1 \cdot \delta_3 = 0.0060$. - As atualizações de bias são $\Delta b_3 = -\eta \cdot \delta_3 = 0.0087$ e $\Delta b_2 = -\eta \cdot a_2 \cdot \delta_2 = 0.0043$, onde $\delta_2 = w_3 \cdot \delta_3 \cdot a_2(1 - a_2)$.

   Nota: As aproximações acima são baseadas na função sigmóide $\sigma(x) = \frac{1}{1+e^{-x}}$ e a taxa de aprendizado $\eta = 0.1$.