

1.

1 / 1 punto

Softmax regression (4 possible outputs)

$$\begin{aligned}
 \times z_1 &= \vec{w}_1 \cdot \vec{x} + b_1 & a_1 &= \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3} + e^{z_4}} \\
 & & &= P(y = 1 | \vec{x}) \quad 0.30 \\
 \circ z_2 &= \vec{w}_2 \cdot \vec{x} + b_2 & a_2 &= \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3} + e^{z_4}} \\
 & & &= P(y = 2 | \vec{x}) \quad 0.20 \\
 \square z_3 &= \vec{w}_3 \cdot \vec{x} + b_3 & a_3 &= \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3} + e^{z_4}} \\
 & & &= P(y = 3 | \vec{x}) \quad 0.15 \\
 \triangle z_4 &= \vec{w}_4 \cdot \vec{x} + b_4 & a_4 &= \frac{e^{z_4}}{e^{z_1} + e^{z_2} + e^{z_3} + e^{z_4}} \\
 & & &= P(y = 4 | \vec{x}) \quad 0.35
 \end{aligned}$$

Para una tarea de clasificación multiclase que tiene 4 salidas posibles, la suma de todas las activaciones suma 1. Para una tarea de clasificación multiclase que tiene 3 salidas posibles, la suma de todas las activaciones debe sumar....

- ☐ Menos que 1
☐ Más de 1
☒ 1
☐ Variará, dependiendo de la entrada x.

**Correcto**

¡Sí! La suma de todas las activaciones de softmax debe sumar 1. Una forma de ver esto es que si $y^{Con_1} = 10, y^{Con_2} = 20, y^{Con_3} = 30$, entonces la suma de $a_1 + a_2 + a_3$ es igual a $\frac{y^{Con_1} + y^{Con_2} + y^{Con_3}}{y^{Con_1} + y^{Con_2} + y^{Con_3}}$ que es 1

2.

1 / 1 punto

Cost

Logistic regression

$$z = \vec{w} \cdot \vec{x} + b$$

$$a_1 = g(z) = \frac{1}{1 + e^{-z}} = P(y = 1|\vec{x})$$

$$a_2 = 1 - a_1 = P(y = 0|\vec{x})$$

$$\text{loss} = -y \log a_1 - (1 - y) \log(1 - a_1)$$

if $y=1$ if $y=0$

$$J(\vec{w}, b) = \text{average loss}$$

Softmax regression

$$a_1 = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + \dots + e^{z_N}} = P(y = 1|\vec{x})$$

$$\vdots$$

$$a_N = \frac{e^{z_N}}{e^{z_1} + e^{z_2} + \dots + e^{z_N}} = P(y = N|\vec{x})$$

Crossentropy loss

$$\text{loss}(a_1, \dots, a_N, y) = \begin{cases} -\log a_1 & \text{if } y = 1 \\ -\log a_2 & \text{if } y = 2 \\ \vdots \\ -\log a_N & \text{if } y = N \end{cases}$$

$\text{loss} = -\log a_j \text{ if } y=j$

Para la clasificación multiclase, la pérdida de entropía cruzada se usa para entrenar el modelo. Si hay 4 clases posibles para la salida, y para un ejemplo de entrenamiento en particular, la verdadera clase del ejemplo es la clase 3 ($y=3$), entonces, ¿a qué se simplifica la pérdida de entropía cruzada?
[Sugerencia: esta pérdida debería disminuir cuando a_3 se hace más grande.]

- ☐ $z_3/(z_1+z_2+z_3+z_4)$
- ☒ $-\log(a_3)$
- ☐ z_3
- ☐ $\frac{-\log(a_1) - \log(a_2) - \log(a_3) - \log(a_4)}{4}$

✓ **Correcto**

Correcto. Cuando la etiqueta verdadera es 3, entonces la pérdida de entropía cruzada para ese ejemplo de entrenamiento es solo el negativo del logaritmo de la activación de la tercera neurona del softmax. Todos los demás términos de la ecuación de pérdida de entropía cruzada ($-\log(a_1)$, $-\log(a_2)$, $-\log(a_4)$) son ignorados

3.

1 / 1 punto

MNIST (more numerically accurate)

```
model    import tensorflow as tf
         from tensorflow.keras import Sequential
         from tensorflow.keras.layers import Dense
         model = Sequential([
             Dense(units=25, activation='relu')
             Dense(units=15, activation='relu')
             Dense(units=10, activation='linear') ])

loss     from tensorflow.keras.losses import
         SparseCategoricalCrossentropy

fit      model.compile(..., loss=SparseCategoricalCrossentropy(from_logits=True) )

predict  logits = model(X)
         f_x = tf.nn.softmax(logits)
```

Para la clasificación multiclase, la forma recomendada de implementar la regresión softmax es establecer `from_logits=True` en la función de pérdida, y también definir la capa de salida del modelo con...

- ☒ una activación 'lineal'
- ☐ una activación 'softmax'



Correcto

¡Sí! Establezca la salida como lineal, porque la función de pérdida maneja el cálculo del softmax con un método más estable numéricamente.