# Softmax Units for Multinoulli Output Distributions

- To generalize to the case with n different values, we similarly use first a linear layer to predict the unnormalized log probabilities

$$\boldsymbol{z} = \boldsymbol{W}^\top \boldsymbol{h} + \boldsymbol{b},$$

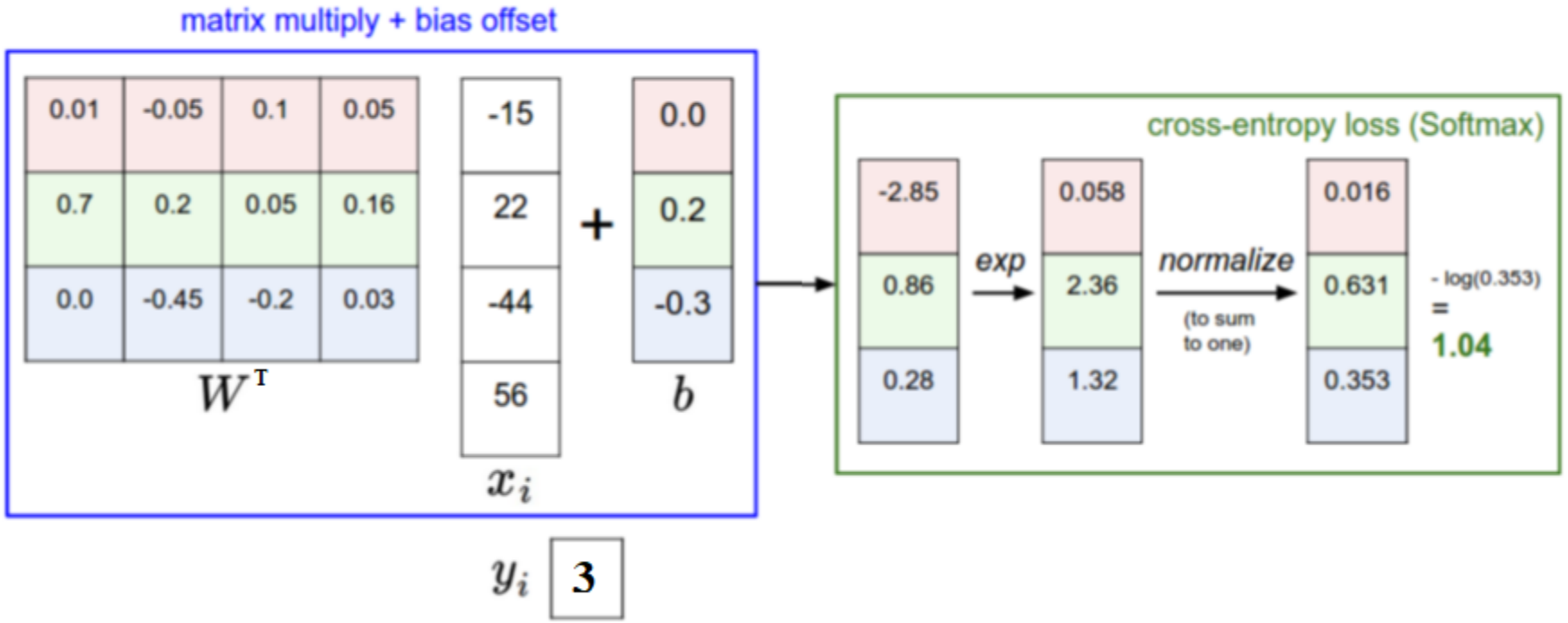
  - We use the softmax function

$$\text{softmax}(\boldsymbol{z})_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}.$$

  - The loss is known as cross entropy loss

$$\log P(\text{y} = i; \boldsymbol{z}) = \log \text{softmax}(\boldsymbol{z})_i = z_i - \log \sum_j \exp(z_j).$$

# Classification Cross Entropy Example




Here we assume the classes are 1, 2, and 3. This sample is from class 3.

# Gradients of the Example – cont.

- The gradients are

$$\frac{\partial L}{\partial z} = \begin{bmatrix} 0.016 \\ 0.631 \\ -0.649 \end{bmatrix}$$

$$\frac{\partial L}{\partial b} = \frac{\partial L}{\partial z} = \begin{bmatrix} 0.016 \\ 0.631 \\ -0.649 \end{bmatrix}$$

$$\frac{\partial L}{\partial W} = \begin{bmatrix} -15 \\ 22 \\ -44 \\ 56 \end{bmatrix} \times \begin{bmatrix} 0.016 \\ 0.631 \\ -0.649 \end{bmatrix}^T = \begin{bmatrix} -0.24 & -9.465 & 9.735 \\ 0.352 & 13.882 & -14.278 \\ -0.704 & -27.764 & 28.556 \\ 0.896 & 35.336 & -36.344 \end{bmatrix}$$

  - You can use the gradients to update the weights and biases