**Entropy**

- In science, entropy is the amount of energy in the system that cannot be harnessed
- In general, it is a **measure of randomness in a system**

**Cross Entropy -** A measure of difference between two probability distributions

**Binary Cross Entropy -** We use it to measure the error of multi-class classification problems

- We can represent the **truth** values (**targets**) as one hot vectors with the right class = 1
    - This represents the probability distribution of the class likelihoods for a data object **x**
- Using this idea, we can compare our **model's prediction y^** to our one hot vector and compute the difference
    - **p -** true one hot vector
    - **q -** predicted probability distribution
    - **log -** ln

$$H(p,q) = -\sum_x p(x)\,\log q(x)$$

- *The above formula assumes that our* ***true y*** *is a one hot vector*
    - If it isn't, we have to use a different formula
- We add a small value **1e-15** to our one hot vectors so that we don't get infinitely large BCE loss because our model didn't account for impossible classes to pop up

**Softmax Function -** used as the **activation function** in the final layer of a **multi-class classification model**

- Squishes all of the output dot products to be 0.0 -> 1.0 and makes sure they all sum up to 1.0
- When implementing a softmax function in code, we subtract a constant (**e.g. max(input)**) for numerical stability
    - This is because it makes the numbers smaller, and dividing two large numbers is numerically unstable
    - *You still get the same result*
    - `np.exp(x - np.max(x))`
      `return e_x / e_x.sum(axis=0)`
        - `(axis=0)` makes sure that it is valid for 2d arrays

$$S(y_i) = \frac{e^{y_i}}{\sum_{j=1}^{j} e^{y_j}}$$

$$\mathbf{h} = \mathbf{w}^T\mathbf{X}$$

$$\text{Logistic regression: } \mathbf{z} = \sigma(\mathbf{h}) = \frac{1}{1+e^{-\mathbf{h}}}$$

$$\text{Cross-entropy loss: } J(\mathbf{w}) = -(\mathbf{y}log(\mathbf{z}) + (1-\mathbf{y})log(1-\mathbf{z}))$$

$$\text{Use chain rule: } \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \frac{\partial J(\mathbf{w})}{\partial \mathbf{z}}\frac{\partial \mathbf{z}}{\partial \mathbf{h}}\frac{\partial \mathbf{h}}{\partial \mathbf{w}}$$

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{z}} = -(\frac{\mathbf{y}}{\mathbf{z}} - \frac{1-\mathbf{y}}{1-\mathbf{z}}) = \frac{\mathbf{z}-\mathbf{y}}{\mathbf{z}(1-\mathbf{z})}$$

$$\frac{\partial \mathbf{z}}{\partial \mathbf{h}} = \mathbf{z}(1-\mathbf{z})$$

$$\frac{\partial \mathbf{h}}{\partial \mathbf{w}} = \mathbf{X}$$

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{X}^T(\mathbf{z}-\mathbf{y})$$

$$\text{Gradient descent: } \mathbf{w} = \mathbf{w} - \alpha\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}$$