

# Cross-Entropy Loss

Marco Willi

## Softmax

```
from matplotlib import pyplot as plt
import numpy as np
import seaborn as sns

np.random.seed(123)
logits = np.random.normal(size=(10, )) * 2
softmax_output = np.exp(logits) / np.exp(logits).sum()

fig, ax = plt.subplots(ncols=2, figsize=(12, 4))
_ = sns.barplot(x=[i for i in range(0, 10)], y=logits, ax=ax[0]).set(
    title="Logits")
_ = sns.barplot(x=[i for i in range(0, 10)], y=softmax_output, ax=ax[1]).set(
    title="Softmax", ylim=(0, 1))
```

## Likelihood

```
import numpy as np

y_true = np.array(
    [0, 1, 0, 1, 1]
)

y_pred = np.array(
    [0.1, 0.8, 0.2, 0.7, 0.9]
)
```

```

# Perfect Prediction
#y_pred = np.array([0, 1, 0, 1, 1])

# Larger Dataset
#y_true = np.repeat(y_true, 10, axis=0)
#y_pred = np.repeat(y_pred, 10, axis=0)

def calculate_likelihood(y_true: float, y_pred: float) -> float:
    return (y_pred ** y_true) * ((1 - y_pred) ** (1 - y_true))

sample_likelihoods = [calculate_likelihood(float(yt), float(yp)) for yt, yp in zip(y_true, y
likelihood = np.prod(sample_likelihoods)
print(f"Total Likelihood: {likelihood:.6f}")

```

## Cross-Entropy

```

from matplotlib import pyplot as plt
from myst_nb import glue
import numpy as np
import seaborn as sns

np.random.seed(123)
k = 5
logits = np.random.normal(size=(5, )) * 2
y_hat = np.exp(logits) / np.exp(logits).sum()
y_true = np.zeros_like(logits)
y_true[1] = 1

fig, ax = plt.subplots(ncols=2, figsize=(12, 4))
_ = sns.barplot(x=[i for i in range(0, k)], y=y_true, ax=ax[0]).set(
    title="True Distribution")
_ = sns.barplot(x=[i for i in range(0, k)], y=y_hat, ax=ax[1]).set(
    title="Predicted Distribution", ylim=(0, 1))

cross_entropy = (- y_true * np.log(y_hat)).sum()
print(f"Cross Entropy: {cross_entropy}")

```