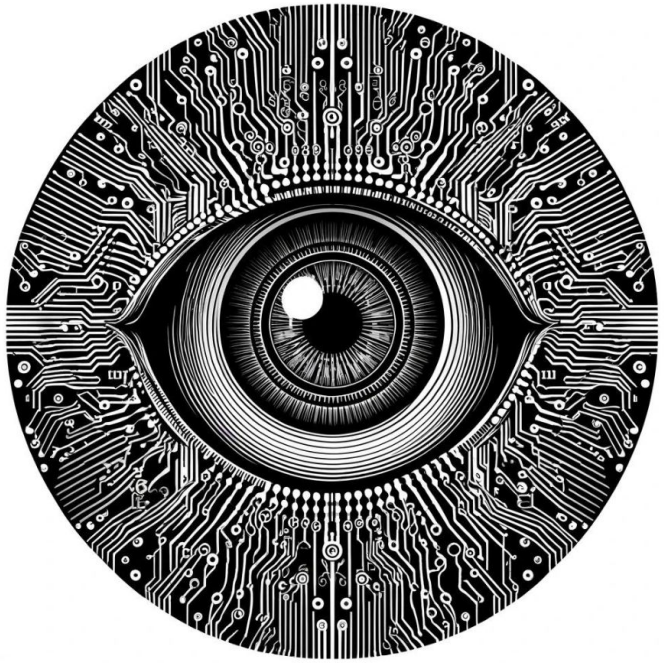
KI Service Zentrum by Hasso-Plattner-Institut HPI

Usage of Binary Cross Entropy Loss



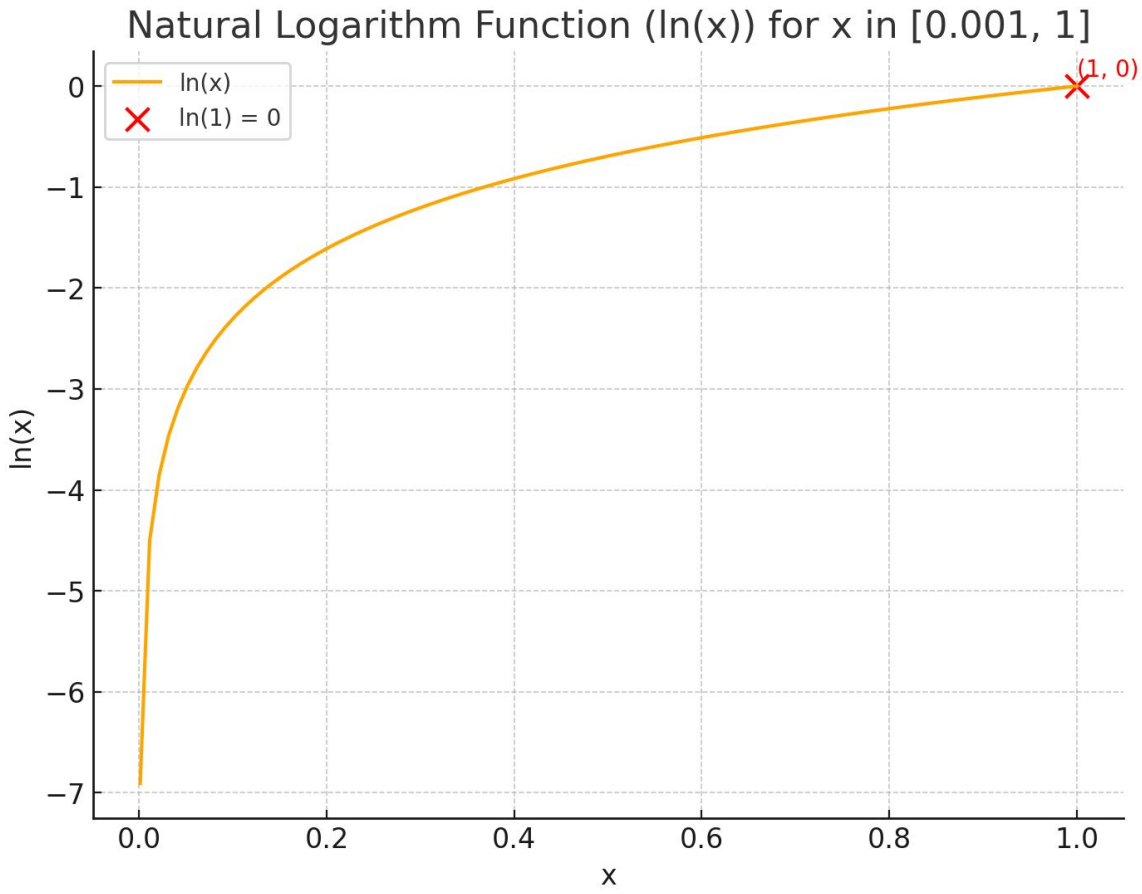
Antonio Rueda-Toicen

SPONSORED BY THE
Federal Ministry of Education and Research

Learning goals

- Encode class labels as probability distributions
- Understand differences between categorical and binary cross entropy loss
- Describe differences between sigmoid and softmax activation functions
- Design network architectures with agreement between the classification task, output layer, activation function, and loss function

The logarithm function and cross entropy



$$H(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

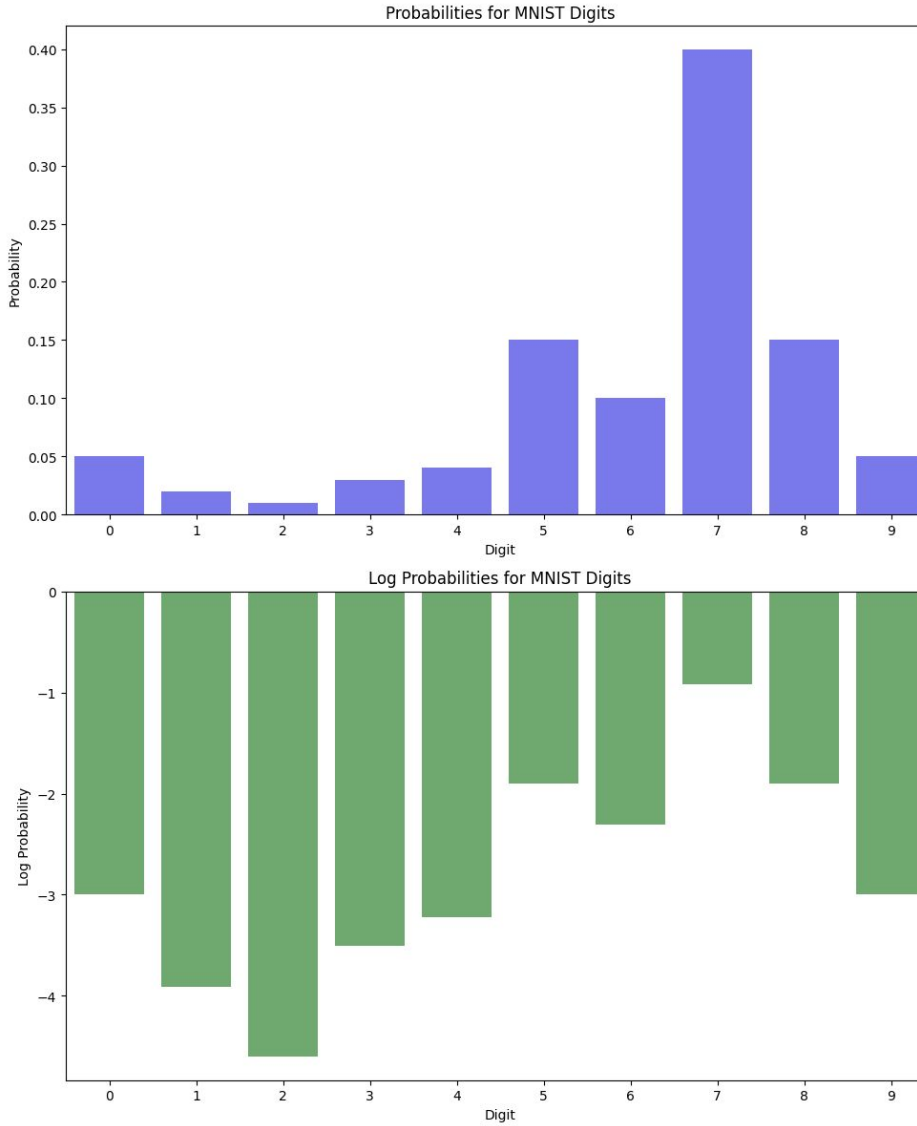
Predicting a single label from multiple classes with a neural network

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)}$$

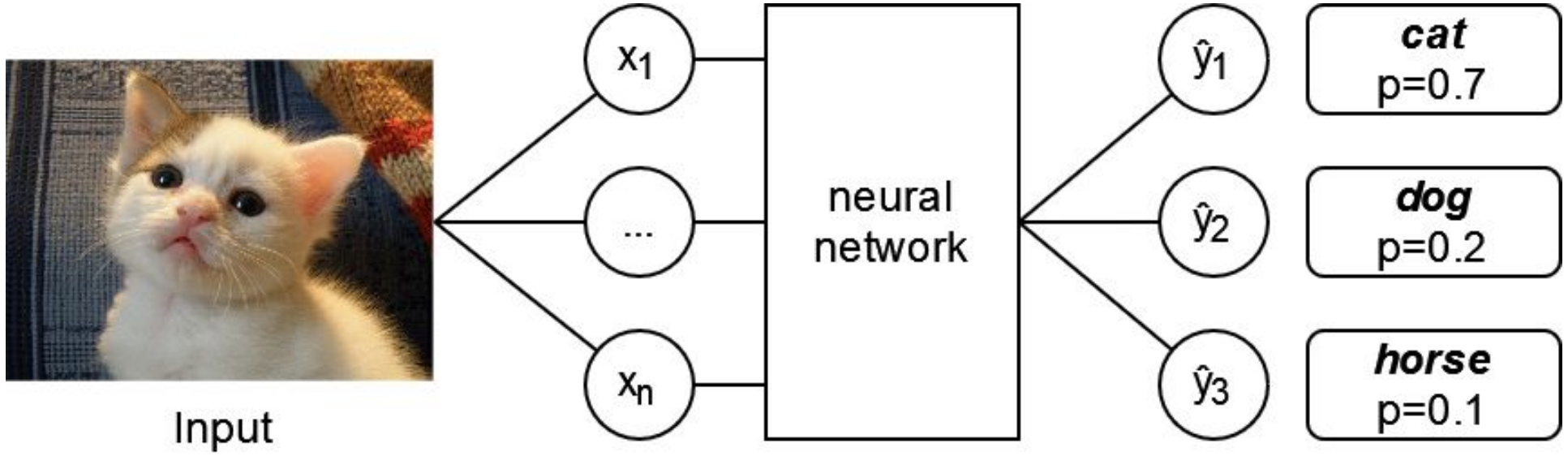
MNIST has a single label per image

$$\log \text{softmax}(x_i) = x_i - \log \sum_{j=1}^{n} \exp(x_j)$$

$$H(y, \hat{y}) = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$



Single label classification: n probabilities for n classes




```
model = nn.Sequential(
    # We get raw logits as output
    nn.Linear(input_size, 3),
)

criterion = nn.CrossEntropyLoss()
loss = criterion(predictions, targets)
```

$$P(\hat{y}_{\text{cat}}) + P(\hat{y}_{\text{dog}}) + P(\hat{y}_{\text{horse}}) = 1$$

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)}$$

Binary Cross Entropy

Used to model events such as

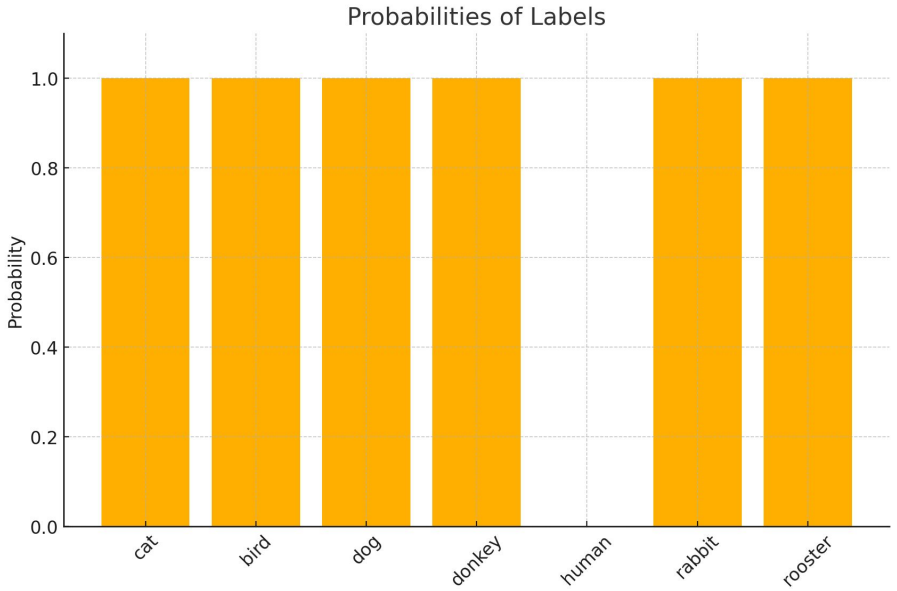
$$P(A) = 1 - P(B)$$

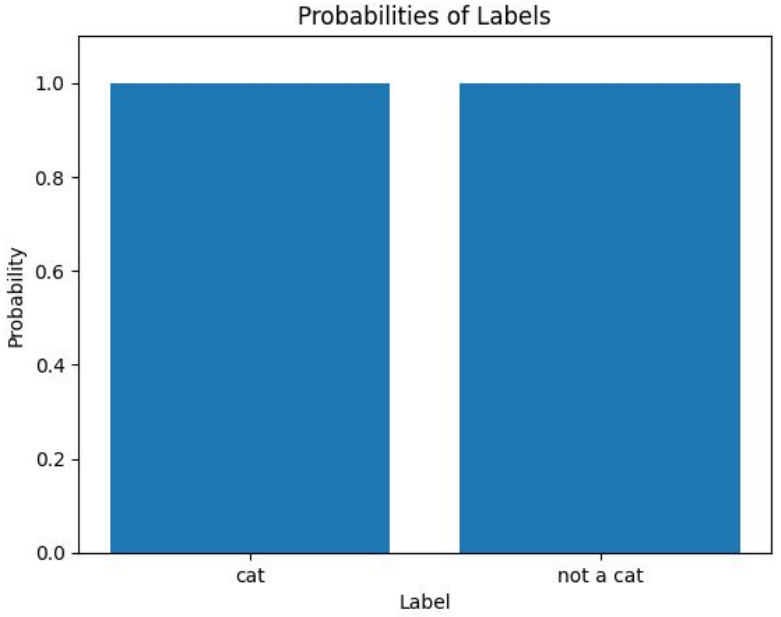
$$\text{BinaryCrossEntropy}(y, \hat{y}) = -\left[y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})\right]$$

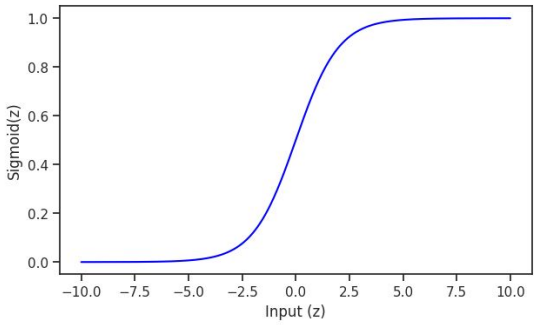
Compare with categorical cross entropy
when having a single class and a single data point

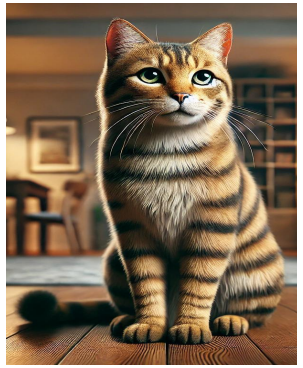
$$H(y, \hat{y}) = -y \log(\hat{y})$$

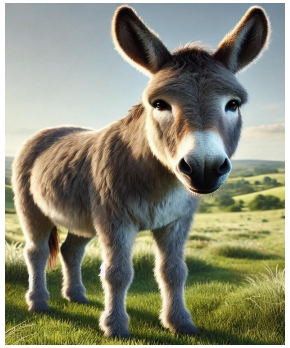
Encoding class labels as probability distributions for binary and multilabel classification



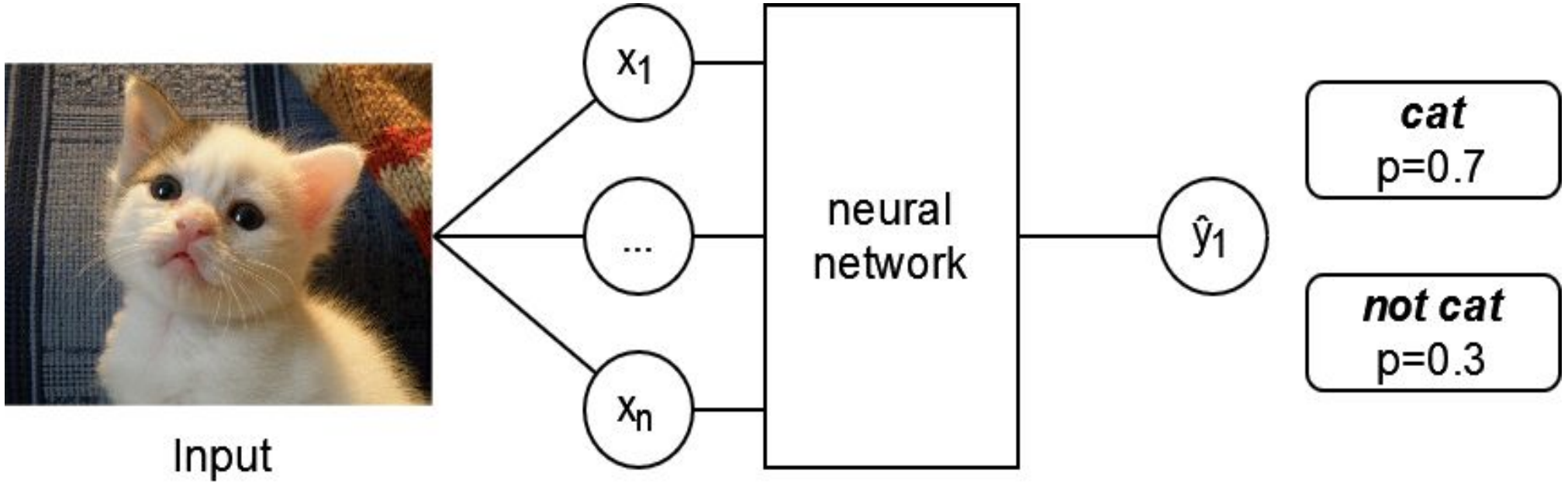




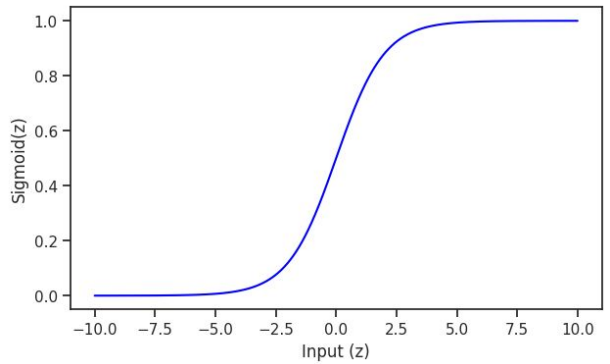




Single label binary classifiers



$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}, \quad \text{where } z = w^T x + b$$



Language-guided detection with Grounding DINO

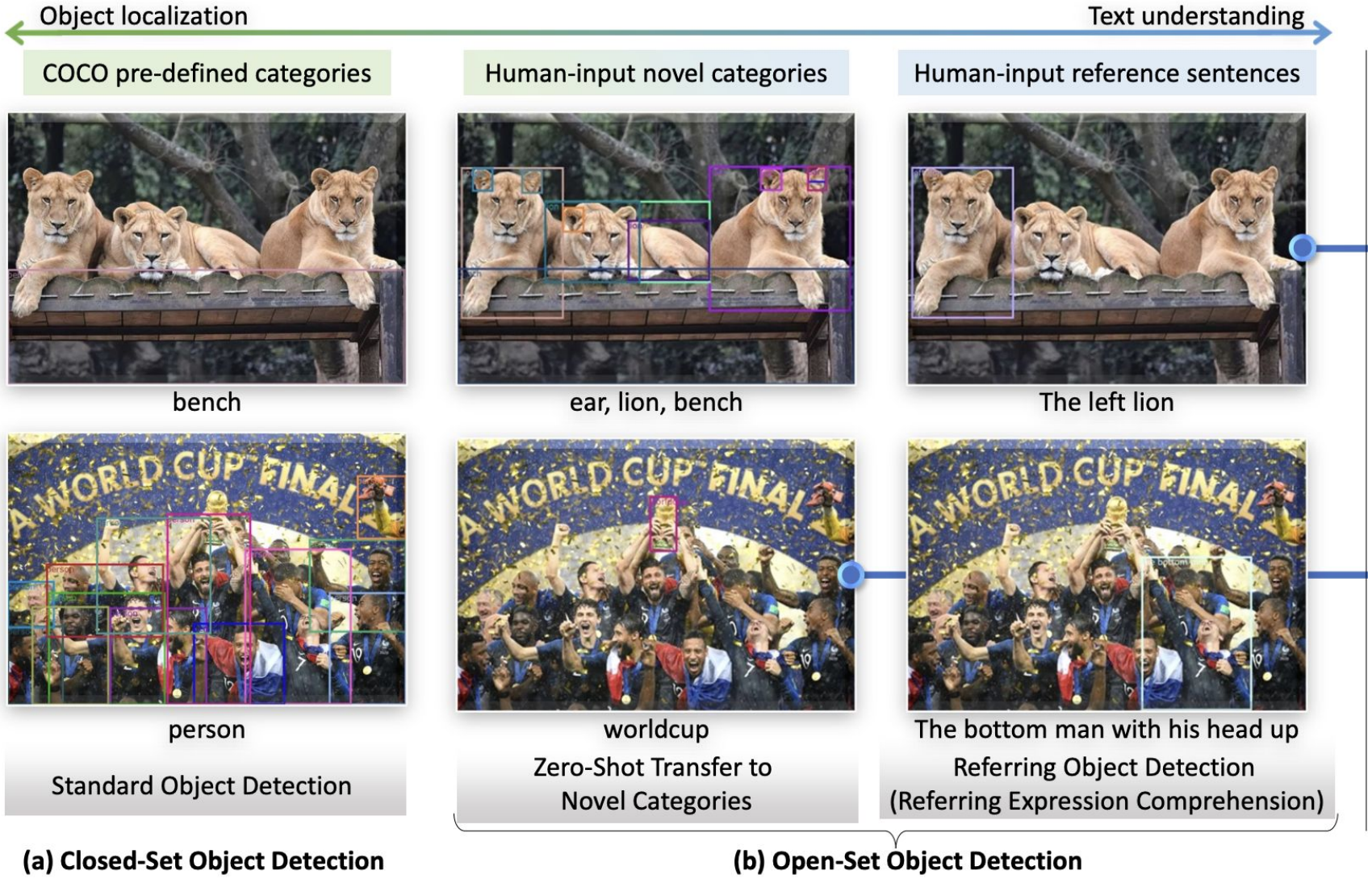
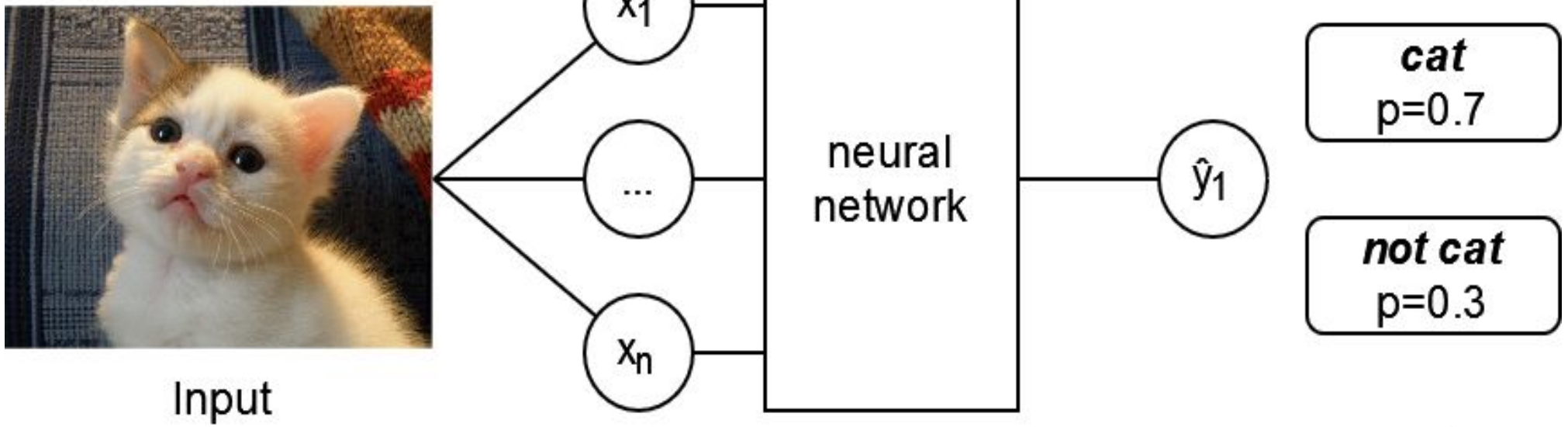


Image from [Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection]

Single label binary classifiers in PyTorch




```
model = nn.Sequential(
     # Raw logits output
    nn.Linear(input_size, 1) )

criterion = nn.BCEWithLogitsLoss()
loss = criterion(logits, targets)
```

```
model = nn.Sequential(
    nn.Linear(input_size, 1),
    # Convert to probabilities
    nn.Sigmoid()
)

criterion = nn.BCELoss()
loss = criterion(predictions, targets)
```

Multilabel classification on satellite images

PLANET · FEATURED PREDICTION COMPETITION · 7 YEARS AGO

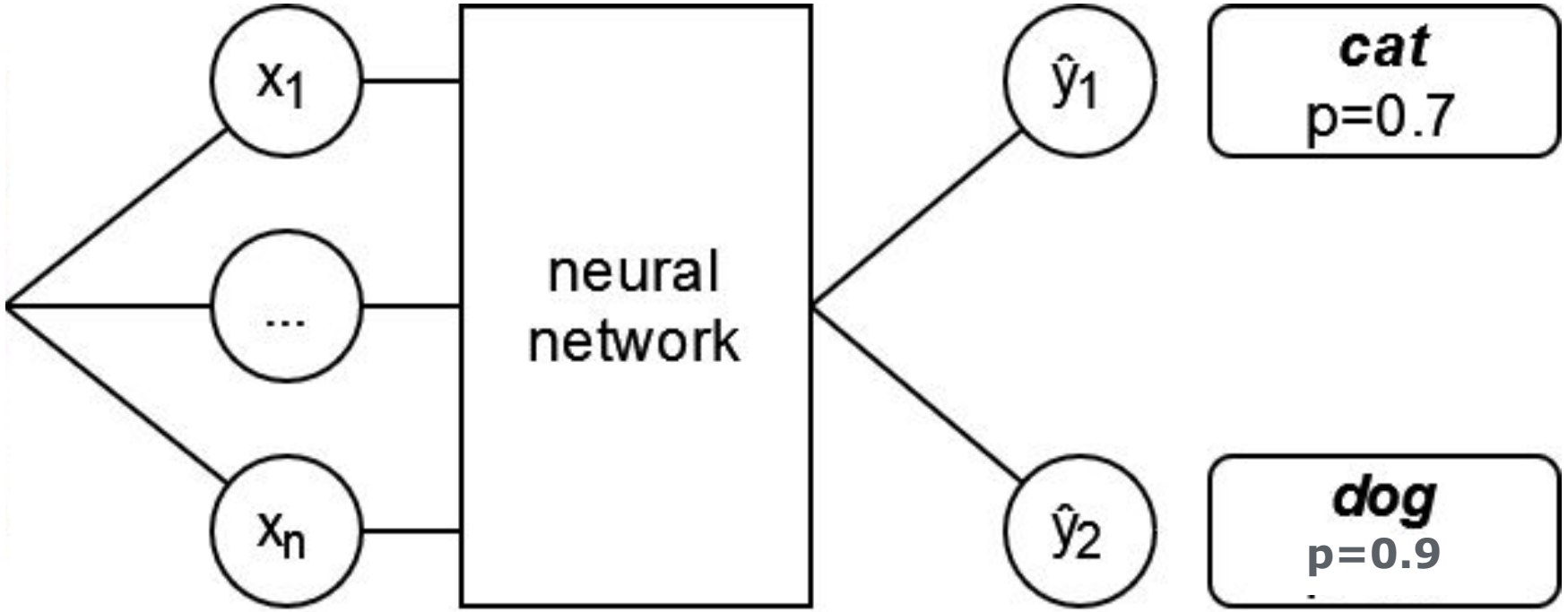
Planet: Understanding the Amazon from Space

Use satellite data to track the human footprint in the Amazon rainforest

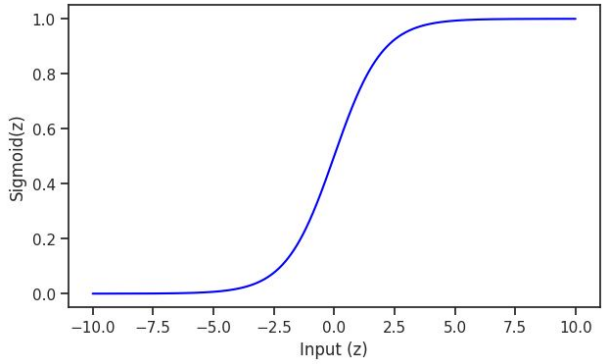
https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/

Multilabel classifiers

Input

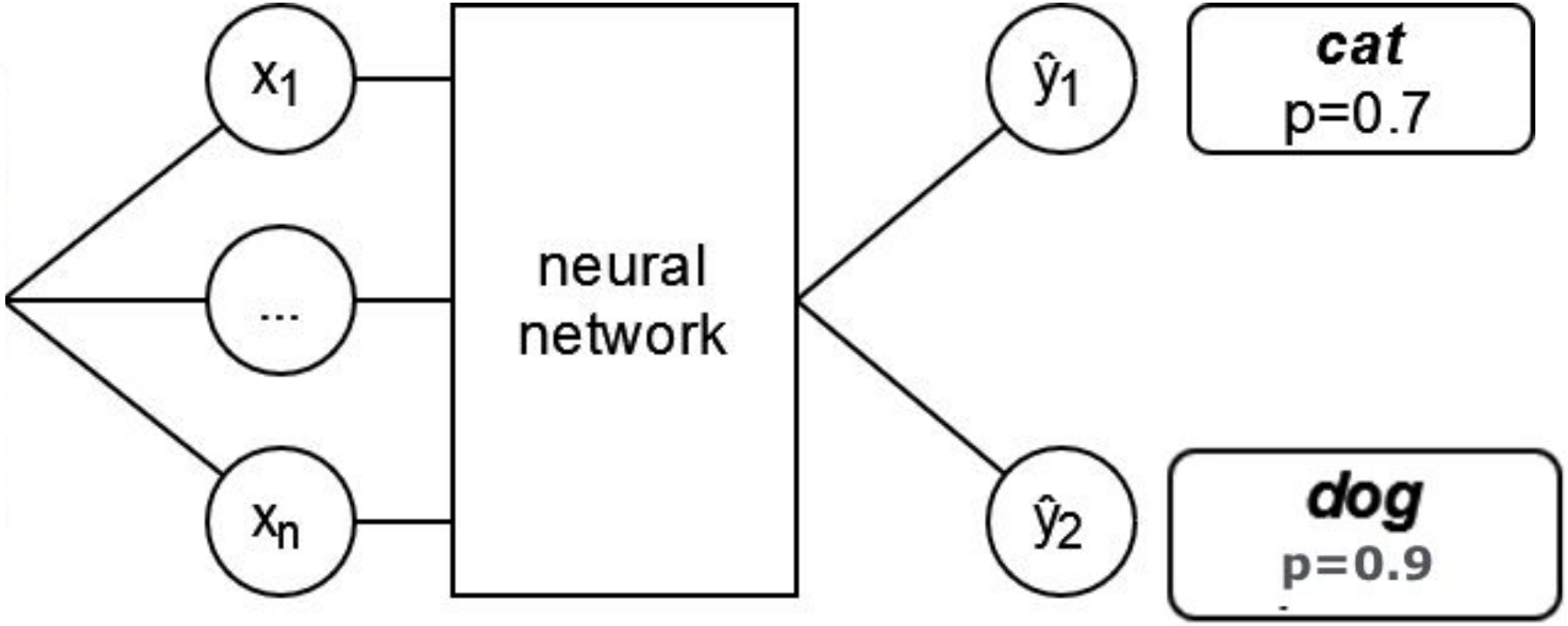


$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}, \quad \text{where } z = w^T x + b$$



Multilabel classifiers in PyTorch

Input



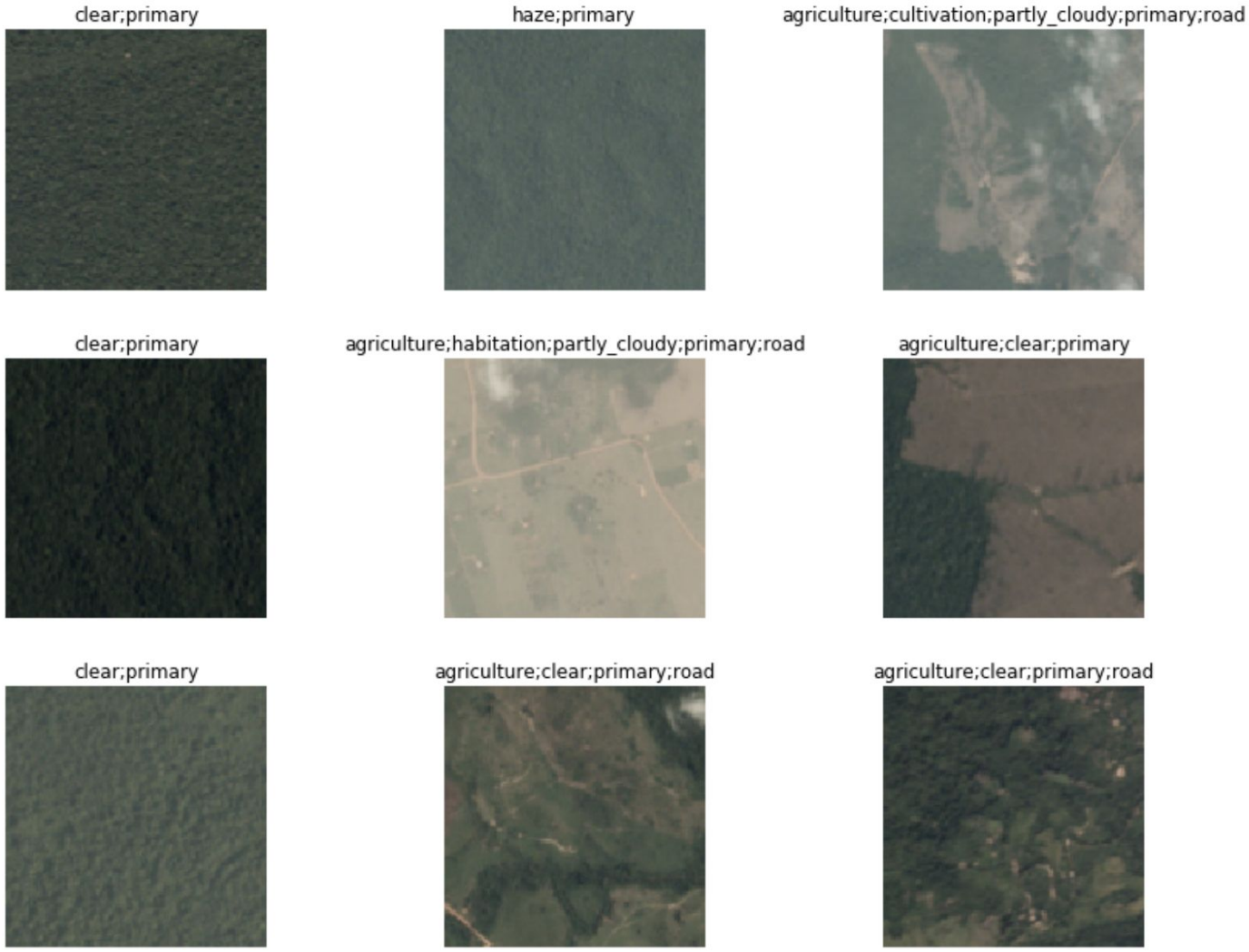

```
model = nn.Sequential(
    nn.Linear(input_size, num_classes),
    # Convert to probabilities
    nn.Sigmoid()
)

criterion = nn.BCELoss()
loss = criterion(predictions, targets)
```

```
model = nn.Sequential(
    # Raw logits
    nn.Linear(input_size, num_classes),

criterion = nn.BCEWithLogitsLoss()
loss = criterion(predictions, targets)
```

Multilabel classification on satellite images



https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/

Summary

Cross entropy and the final activation function

- Cross entropy measures the mismatch between predicted and true distributions
- Softmax handles single-label problems with multiple classes
- Sigmoid handles both binary and multi-label cases

Network design

- Match output layer size to number of classes
- Consider single-label vs multi-label requirements for final layer
- Choose activation function based on classification type (e.g. single label vs multi-label)

Different tasks require different variants of cross entropy loss

- Use `nn.BCELoss` or nn.`BCEWithLogitsLoss` for binary or multi-label classification tasks
- Use `nn.CrossEntropyLoss` for single label tasks

SPONSORED BY THE
Federal Ministry of Education and Research

Further reading

LogSoftmax vs Softmax

- https://discuss.pytorch.org/t/logsoftmax-vs-softmax/21386

SPONSORED BY THE
Federal Ministry of Education and Research