Machine Learning 410

Lesson 0

Review of Multi-Class Classification

Steve Elston

Background for Deep Learning

- Review of multi-class distributions
- Tensors for deep learning
- Basics of image data
- Introduction to Keras

Review of Binary Classification

Bernoulli distribution for probability of success

Observation: $\nu = 1$:

$$p(\nu = 1) = \Theta$$

$$where$$

$$\nu = an\ observation$$

$$\Theta = probability\ parameter$$

Review of Binary Classification

We extend the Bernoulli distribution for multiple trials with the **Binomial distribution** for k successes in n trials:

$$p(\nu = k \mid \Theta) = \binom{n}{k} \Theta^k (1 - \Theta)^{n-k}$$

Were the Binomial coefficient, pronounced n choose k is:

$$\binom{n}{k}$$

Review of Binary Classification

How do perform classification with the Bernoulli distribution?

Use the **logistic** or **sigmoid** function

$$\sigma(x) = \frac{L}{1 + e^{-k(x_0 - x)}}$$

$$where$$

$$L = max\ value$$

$$k = slope$$

$$x_0 = sigmoid\ midpoint$$

Review of Binary Classification

Simplify the logistic function if k = 1, L = 1 and x_o = 0:

$$\sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{1 + e^x}$$

Classification with the Categorical Distribution

- What are the distribution for multi-class problem?
- Use the **Categorical distribution** with probability mass function:

$$p(x = i \mid \boldsymbol{\theta}) = \theta_i$$

Where: $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$

And: $\sum_{i=1}^{k} \theta_i = 1.0$

Classification with the Categorical Distribution

- How do we create a categorical classifier?
- Use a softmax function:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}$$

The normalization, $\sum_{k=1}^{K} e^{z_k}$, ensures the probabilities sum to 1.0

Classification with the Categorical Distribution

- What is the output of softmax?
- One value for each category
  - For example, if we have 10 categories, there are 10 output values
  - Take the max as the most probable category
- Label must be one-hot encoded
  - Binary value for each possible category
  - Only one 1, others 0

Tensors for Deep Learning

For deep learning models we need a way to represent high dimensional values

- Input data
- Model parameters
- Transformations
- Output

Tensors are a generalization of multi-dimensional arrays

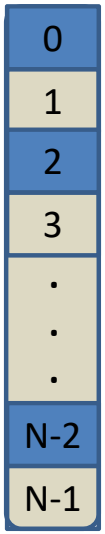
Not to be confused with tensors in physics and engineering

Tensors for Deep Learning

Zero dimensional tensor = scalar

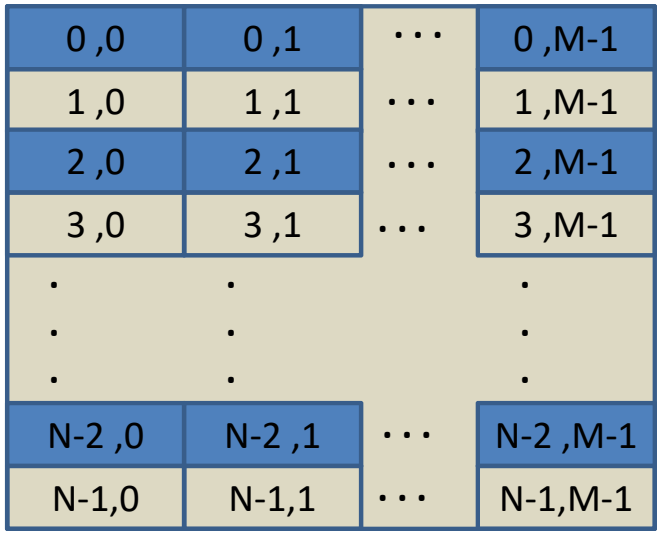
Tensors for Deep Learning

One dimensional tensor = vector



Tensors for Deep Learning

Two dimensional tensor = matrix



Working with Image Data

Deep learning has revolutionized image analysis and understanding

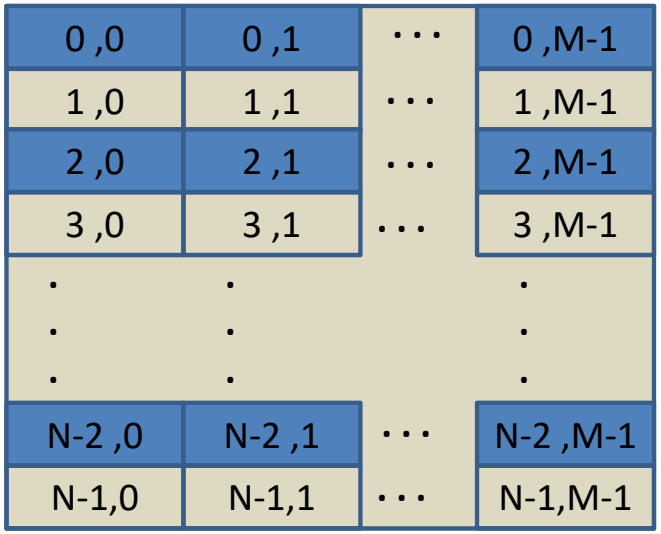
Common types of image data:

- Grey scale
- Color images
- Color video

Image data is represented by tensors

Working with Image Data

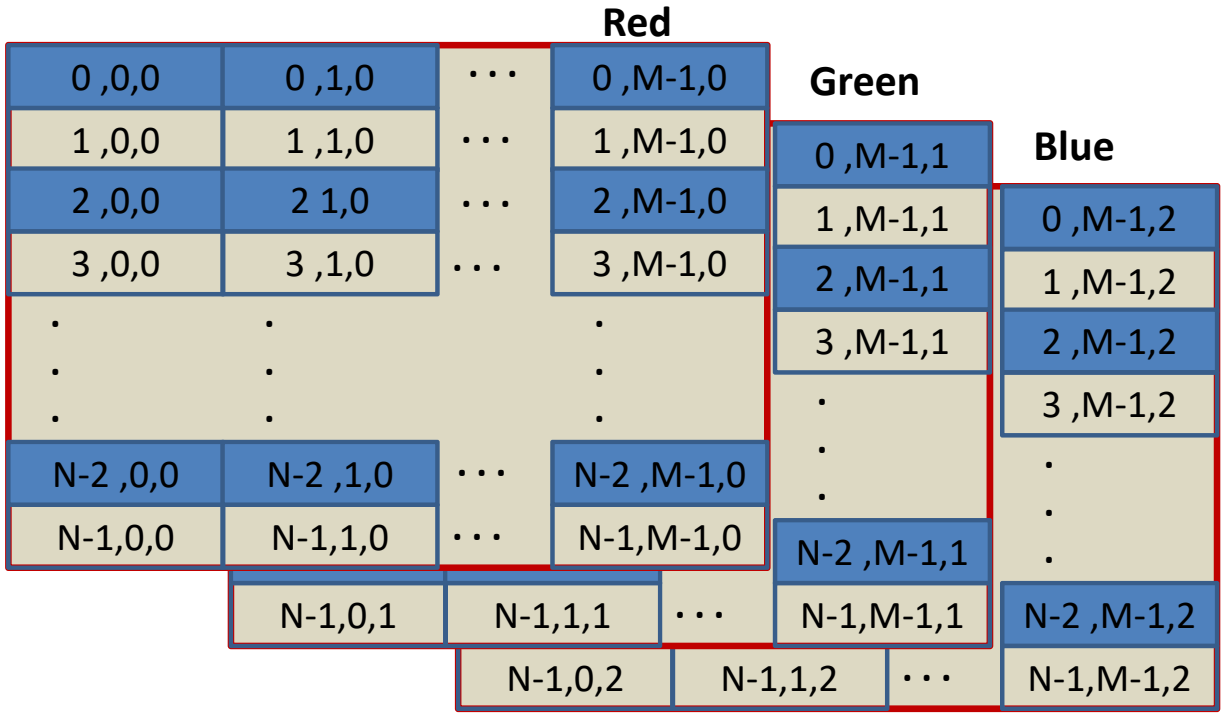
A grey scale image is represented as a two dimensional tensor of pixel values



Pixel values have range {0-255}; white-black

Working with Image Data

A color image is a 3-dimensional tensor representing the three color channels



Intensity of color in each channel is on a scale of {0-255}

Working with Image Data

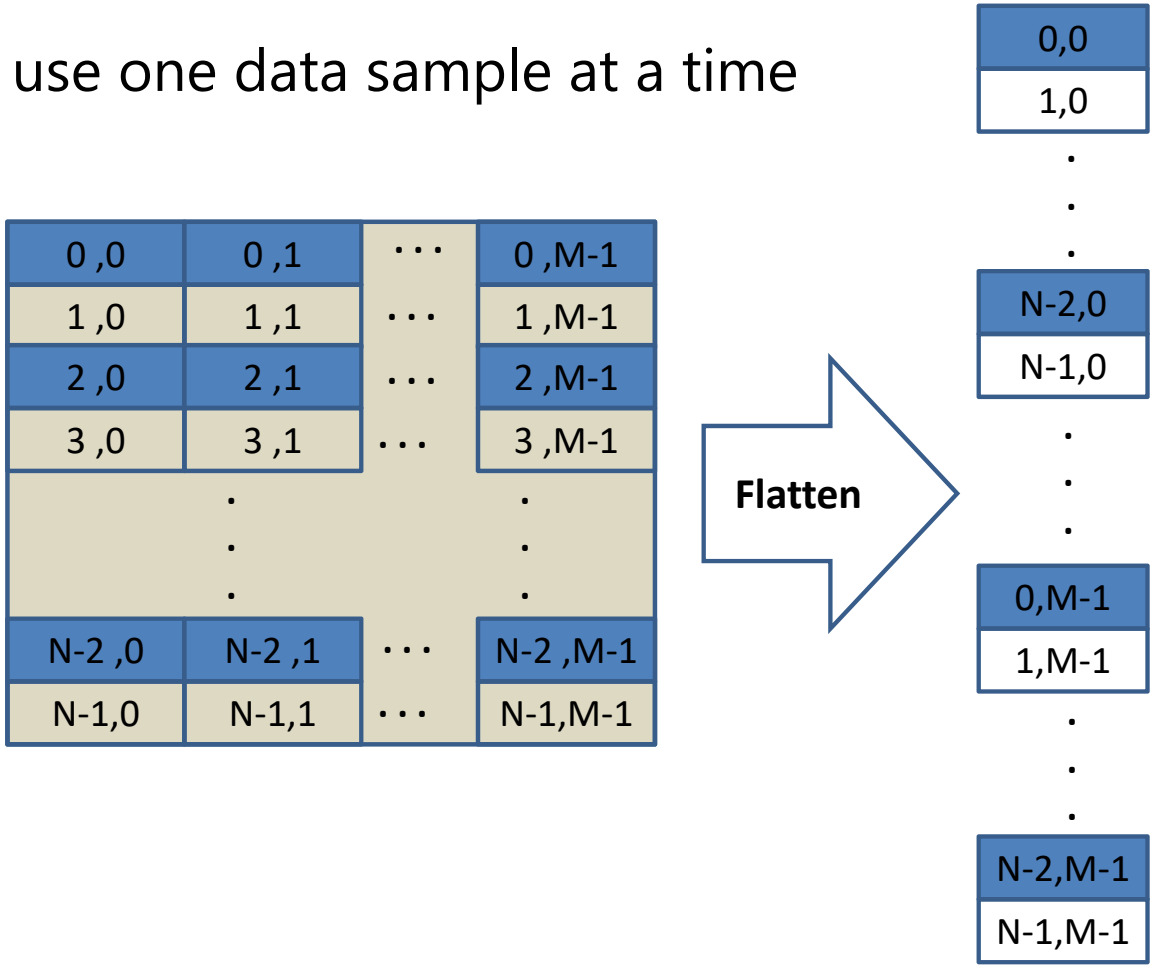
Video is represented as a 4-dimensional tensor

- 2 dimensions for each color channel
- 3rd dimension is color channel
- 4th dimension is the time sequence of images

Working with Image Data

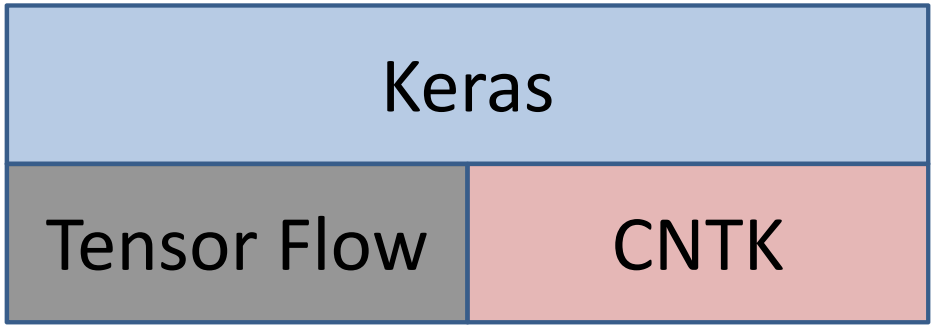
Machine learning algorithms use one data sample at a time

For image data, we must **flatten** the image:



Introduction to Keras

- Keras abstracts common deep learning model definitions and operations
- Use on top of deep learning framework:



- Extensive documentation at www.keras.io