12.06.25

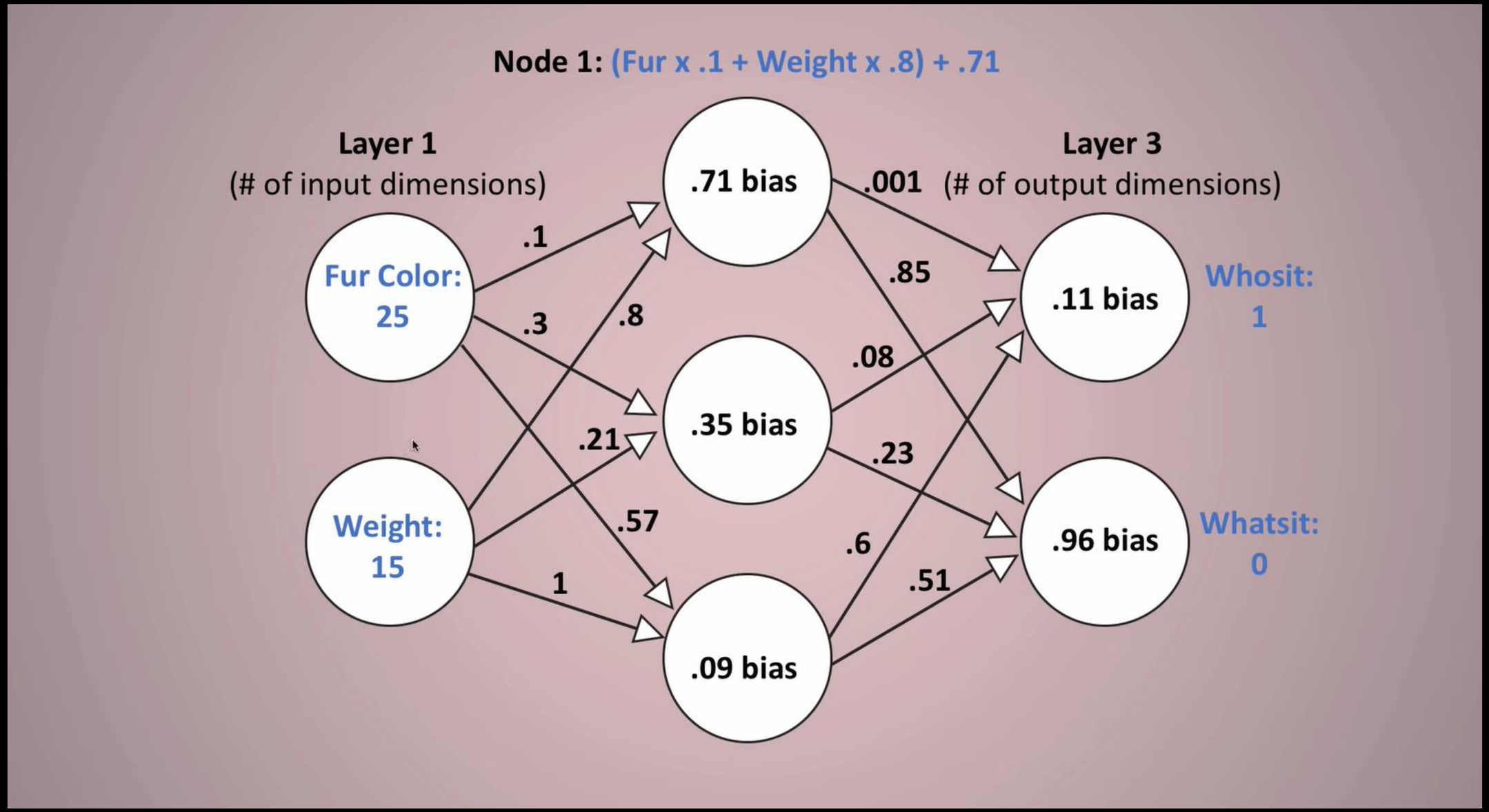
OPENCAMPUS.sh

# Introduction to Data Science and Machine Learning

# NEURAL NETS

- Quiz
- Frameworks for Implementing Neural Networks (NN)
- Implementing a NN with TensorFlow
- Additional Layer Types for NNs

# QUIZ




LearnCode.academy. (2018, June 11). *Neural Networks Explained—Machine Learning Tutorial for Beginners*. https://www.youtube.com/watch?v=GvQwE2OhL8I

# Learning Rate:

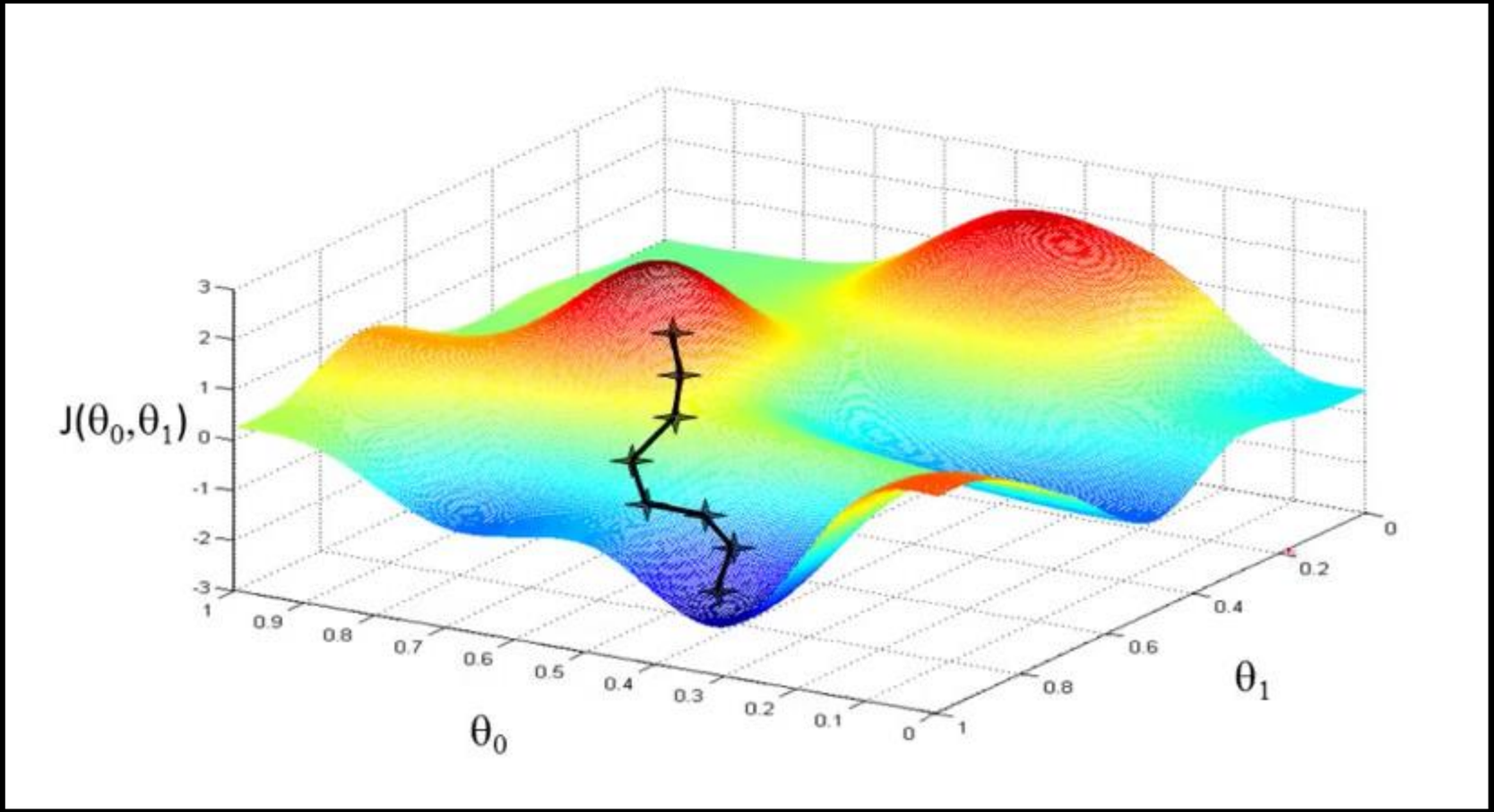
**How much should this step outcome affect our weights and biases?**

# Momentum:

**How much should past outcomes affect our weights and biases?**

**change = (learningRate * delta * value) + (momentum * pastChange)**

# OPTIMIZER PARAMETERS




- **Step size for approaching the cost minimum ("Learning Rate")**
- **Resistance to direction changes ("Momentum")**

Quelle: https://www.coursera.org/learn/machine-learning

# PARAMETERS OF THE “ADAM” OPTIMIZER

- **Learning parameter for optimization:**
  **alpha (learning rate)**

- **Proportion of current gradient in calculating the next optimization step:**
  - **beta1 (decay rate for the direction) and**
  - **beta2 (decay rate for the magnitude of gradients)**

# HYPERPARAMATERS IN NEURAL NETS

- **Architecture Definition:**
  - **Number of hidden layers in the network**
  - **Types of hidden layers**
  - **Number of neurons per hidden layer**
  - **Selection of activation function**
- **Selection of cost function („Loss Function“)**
- **Selection of the optimizer**
- **Setting of the learning rate for the optimizer**

# LIBRARIES FOR NEURAL NETS

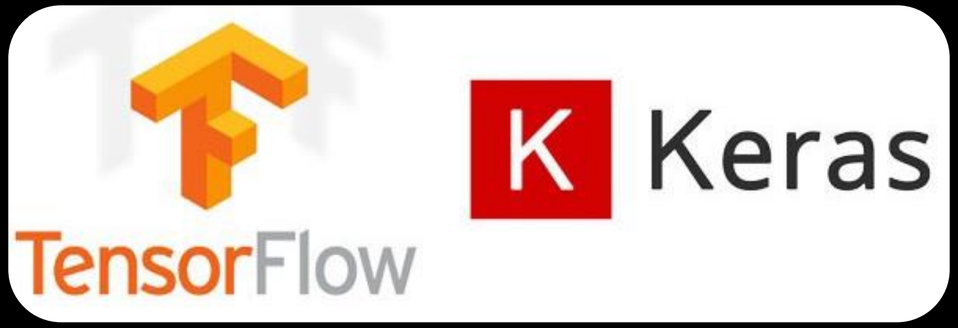

K
Keras


TensorFlow


scikit
learn


PYTORCH


Transformers




- **TensorFlow 0.1 (Nov 2015):** Released as open source software by Google; developed by the Google Brain Team for internal research and production)
- **TensorFlow 1.4 (Nov 2017):** Development of the Keras API as a high-level API for TensorFlow and other ML libraries, to increase user-friendliness for commonly used models.
- **TensorFlow 2.0 (Sep 2019):** Keras is integrated as a high-level API into TensorFlow.
- **TensorFlow 2.3 (October 2020):** Significant performance improvements, distributed training, quantized training, and improved mobile deployments.
- **Keras 3.0 (Dec 2023):** Major release that extends support for multiple backends, including TensorFlow, JAX, and PyTorch, making Keras a versatile framework for various deep learning needs.

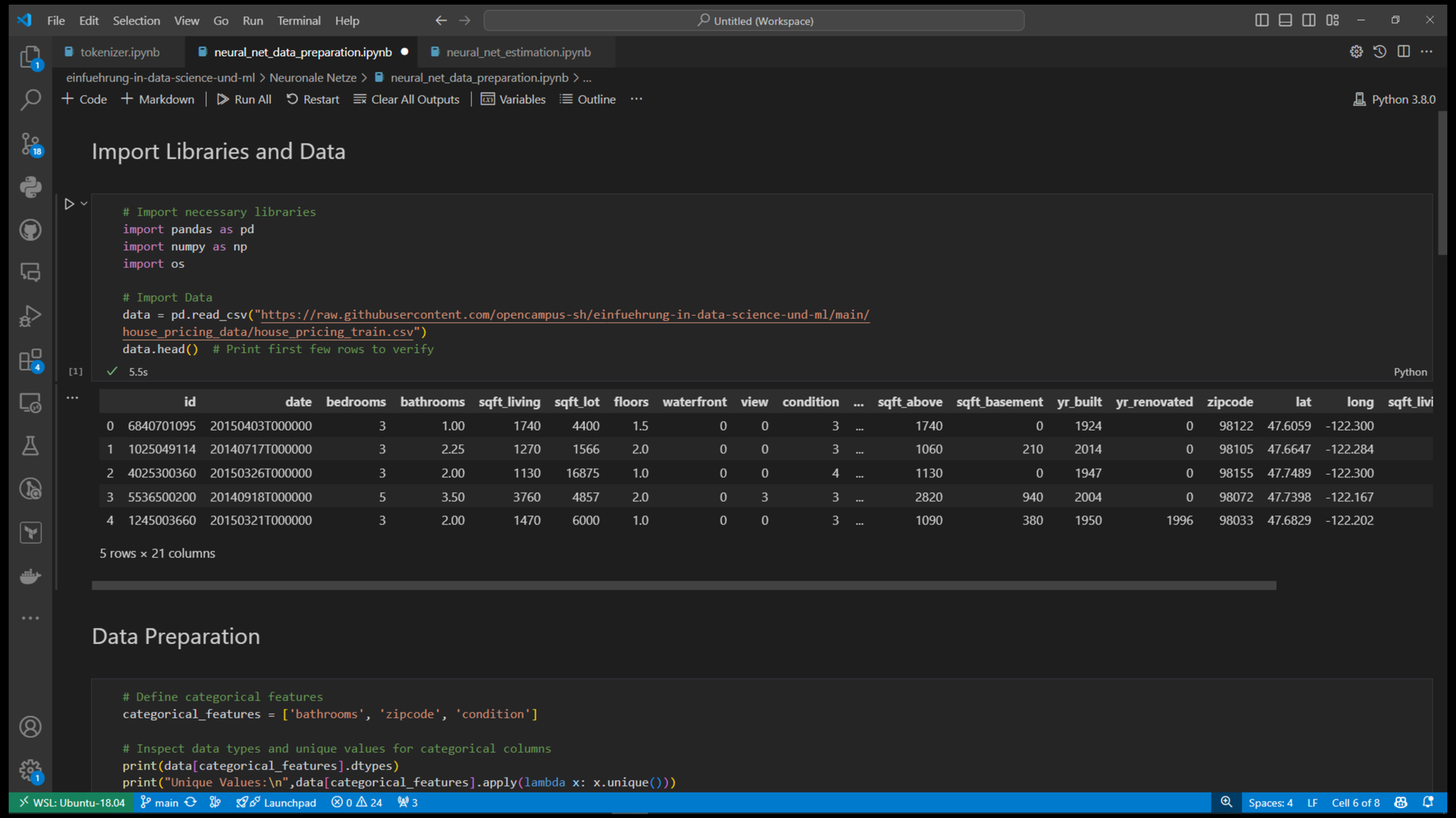

- **Transformers 1.0 (December 2018):** Hugging Face releases the first version of the Transformer library as open source with implementations of popular transformer models like BERT, GPT, and others.
- **Transformers 2.0 (March 2020):** Support for fine-tuning, production deployment, and quantization. Higher performance and more intuitive APIs.
- **Transformers 3.0 (November 2020):** Support for more task areas like computer vision, audio, and reinforcement learning. Performance optimizations.
- **Transformers 4.0 (May 2022):** Improved performance and integration of additional new techniques for enhancing model optimization.

# DATA PREPARATION

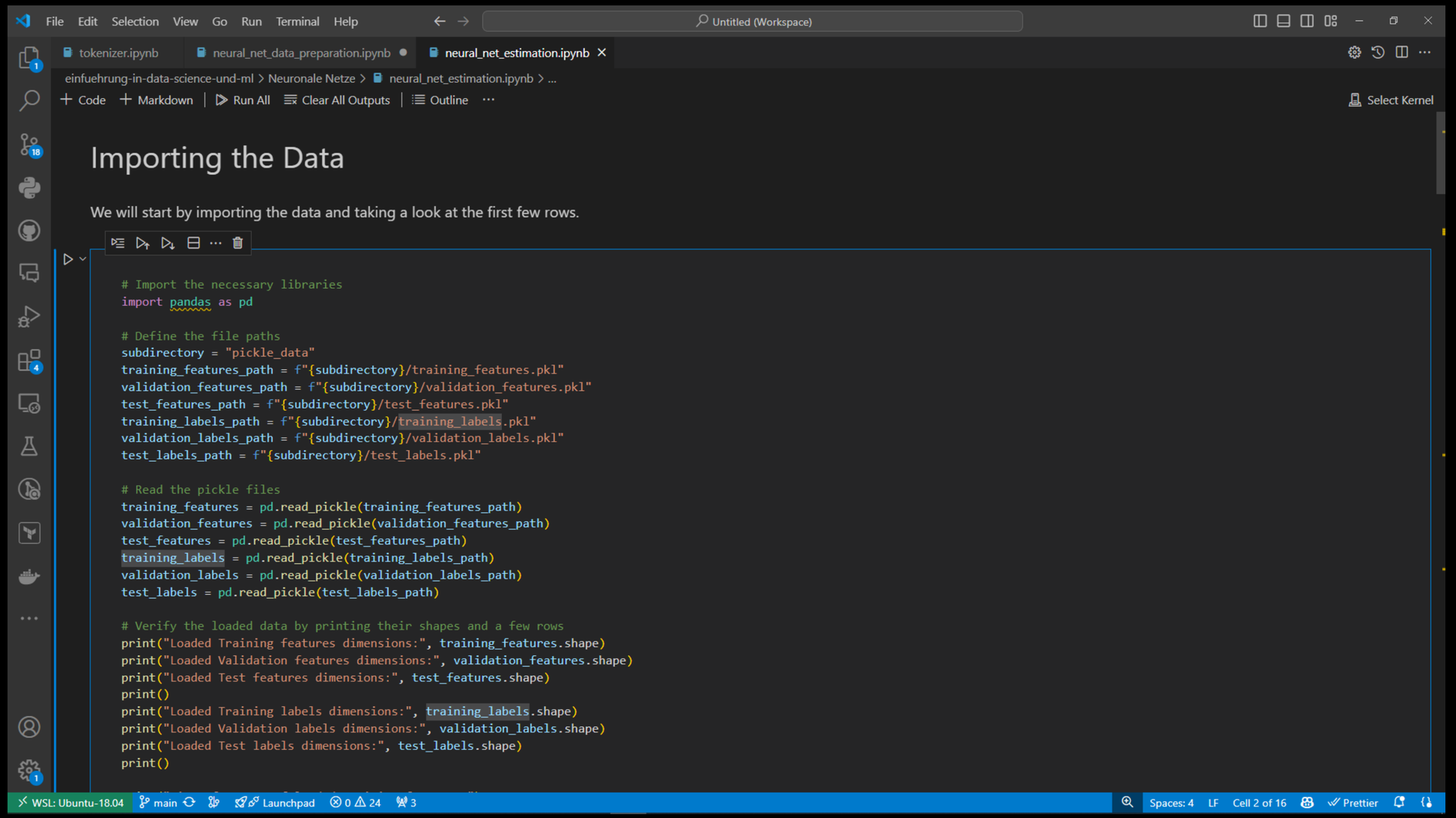
**For every modeling, the data must have the following properties:**

1. **There must be no missing values.**
2. **All values must be numbers.**
3. **Categorical variables are one-hot encoded.**

# EXAMPLE DATA PREPARATION FOR THE TRAINING OF A NEURAL NET



# EXAMPLE DEFINITION AND OPTIMIZATION OF A NEURAL NET



# BREAKOUT

- Load the example notebooks from below into your Codespace and run them once unchanged.

- Supplement your data preparation with the steps performed in this example notebook:
  1) One-hot encoding of categorical variables
  2) Removing cases with missing values
  3) Export of training and validation data as pickle files

- Estimate a first neural network based on this example notebook.

# BATCHES, STEPS AND EPOCHS

***Batch***

- **The entire set of training data is divided into separate subgroups of equal size.**
- **The standard batch size in TensorFlow is 32.**

***Step***

- **A single iteration of gradient descent performed on one batch of data, during which all model weights are updated once.**

***Epoch***

- **Optimization of the model using the complete training data:**
  **Number of Steps × Batch Size = Training Sample Size**
- **Depending on the model, very few epochs may suffice, or several hundred or thousand may be needed for optimization.**

# NORMALIZATION

**Definition:**

- **Subtracting the mean and dividing by the standard deviation.**

**Ensures all input features are on similar scales, which stabilizes training and speeds up convergence.**

# BATCH NORMALIZATION

- Performing normalization at the batch level

Additional optimization parameters:

- Exactly identical means and standard deviations are not necessarily optimal for modeling purposes

→ Normalization parameters are incorporated as trainable parameters

# LEARNING RESSOURCES

- Watch this video (7 minutes) to better understand the properties of dropout layers.

- Watch this video (5 minutes) to better understand the benefits of normalization.

- Complete the first chapter of this course on DataCamp to learn about identifying missing values.

# TASKS

- Examine all your model variables for the existence of missing and implausible values.

- Train a first neural network for your dataset.
  (Delete all rows with missing values.)