JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

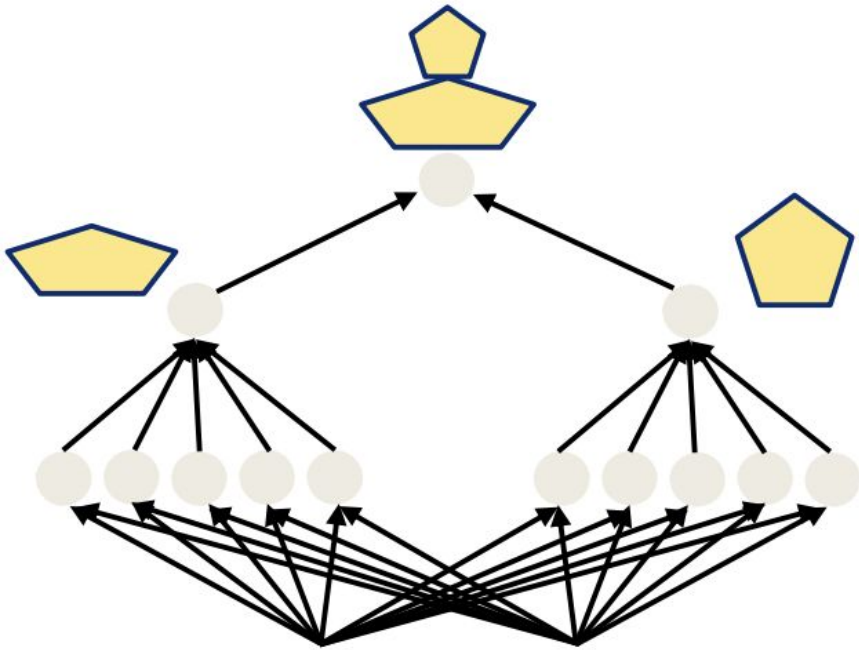
Intro Homework 4

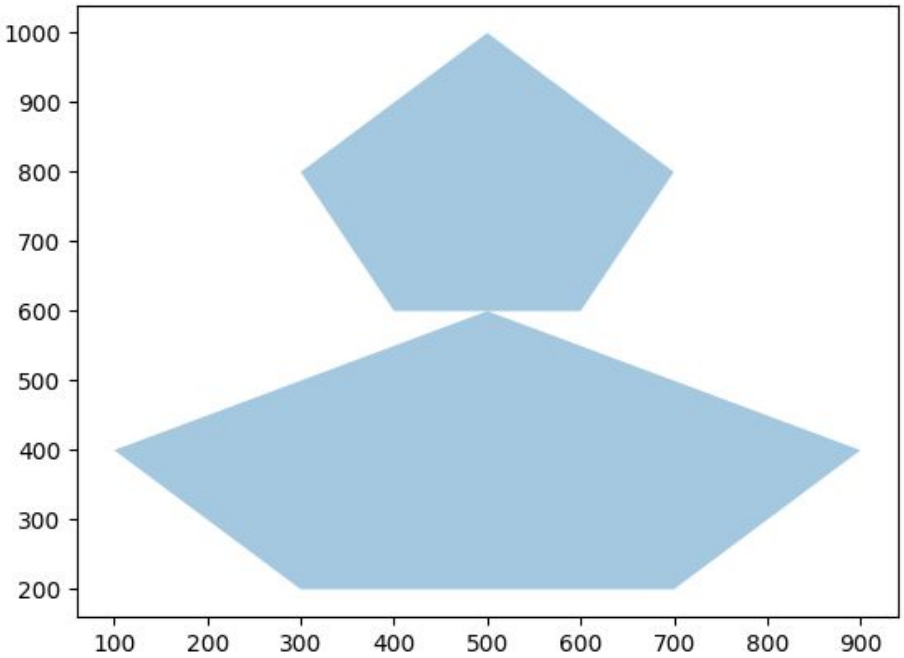
Homework 4 Walkthrough

- Programming assignment, due Oct 11.
- GradeScope submission guideline
  - Homework 4 – report
    - This should include your answers/discussion to each question, and all the plots generated in the programming part
  - Homework 4 – notebook
    - Notebook file (ipynb file)
    - Exported PDF version of your notebook with all cell outputs.
    - hw4_utils.py

Homework 4 Walkthrough

- **Q1 – MLP for polygon classification**





Homework 4 Walkthrough

- **Q1 – MLP for polygon classification**

(a) Implement the “AND gate” and “OR gate” and the “unit step function” to predict all points within the 2 polygons as the positive class. Use the `predict_output_v1()` to test your implementations.

Modify the code in `predict_output_v2()` to predict only the points in the first polygon as the positive class. Attach the visualization to your report.

Homework 4 Walkthrough

- **Q1 – MLP for polygon classification**

(b) Build an **MLP** using sigmoid activation functions to replace the AND/OR gates. Implement the function **preprocess_data**. Implement the **training** function with mini-batch support. Implement visualization functions for your results.

Homework 4 Walkthrough

- **Q1 – MLP for polygon classification**

Notes:

- preprocess_data:
  - Try figure out what preprocessing is needed for the dataset (or check out the discussions on piazza).
  - This has HUGE impact on your training results
- train():
  - Note that seed_value should be used
- Results:
  - Train 5 different models under the five seeds. Calculate the mean/stdev of the results from the 5 runs
  - provide a polygon visualization figure for each of the trained models

Homework 4 Walkthrough

- **Q1 – MLP for polygon classification**

(c) Build an MLP with a larger capacity (increase the depth and width). You may increase the epoch number if you observe that your model does not converge by 500 epochs due to the additional parameters.

Notes:

- You only need to design one new architecture with increased depth AND width. And run it for 5 runs. No other hyperparameters should be changed
- The result should be at least as good as 1(b)

Homework 4 Walkthrough

- **Q1 – MLP for polygon classification**

(c) Build an MLP with a larger capacity (increase the depth and width). You may increase the epoch number if you observe that your model does not converge by 500 epochs due to the additional parameters.

Notes:

- You only need to design one new architecture with increased depth AND width. And run it for 5 runs. No other hyperparameters should be changed
- The result should be at least as good as 1(b)

Homework 4 Walkthrough

- **Q2 – FashionMNIST**

In this problem, we will explore convolutional neural networks (CNNs). We will be working with the FashionMNIST dataset.This dataset only has a train-test split, therefore we will be using the last 10000 training instances as the validation set.

- Design a small CNN (e.g. **3-4 hidden layers**) using (1) convolution and pooling layers (2) activation functions (other than sigmoid)
- Try to improve your model's performance by including additional architecture elements. (1) dropout (2) batch normalization (3) data augmentation (4) different optimizers

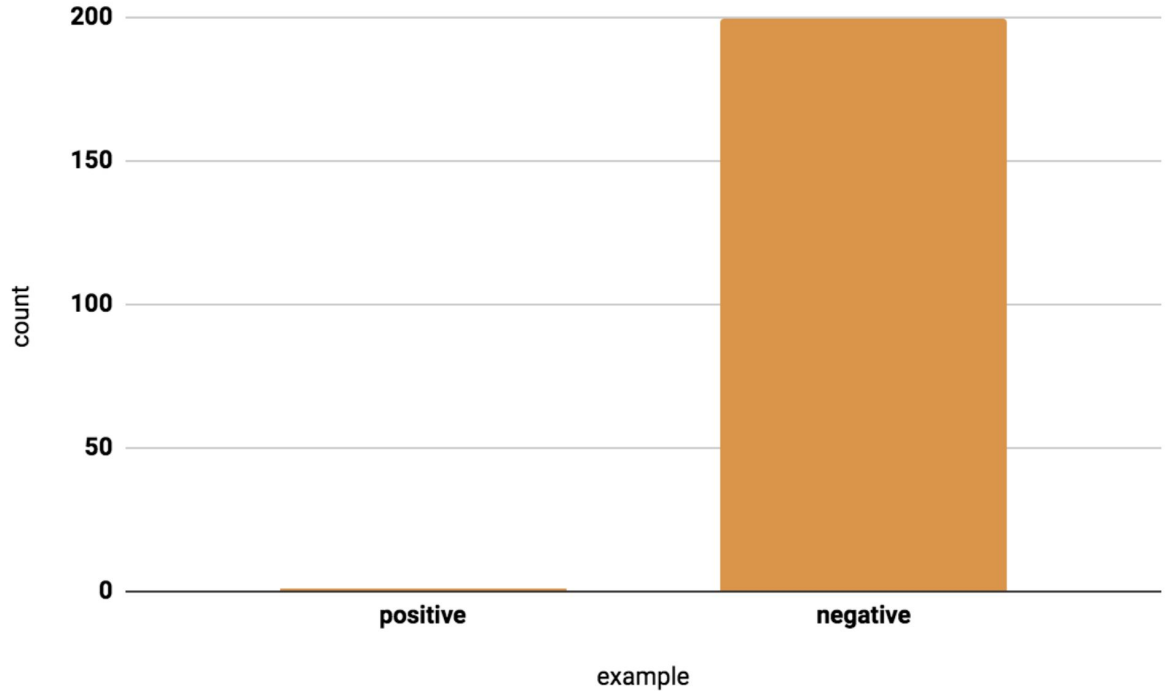
JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Data!

Ping-Cheng Ku

Understanding the data

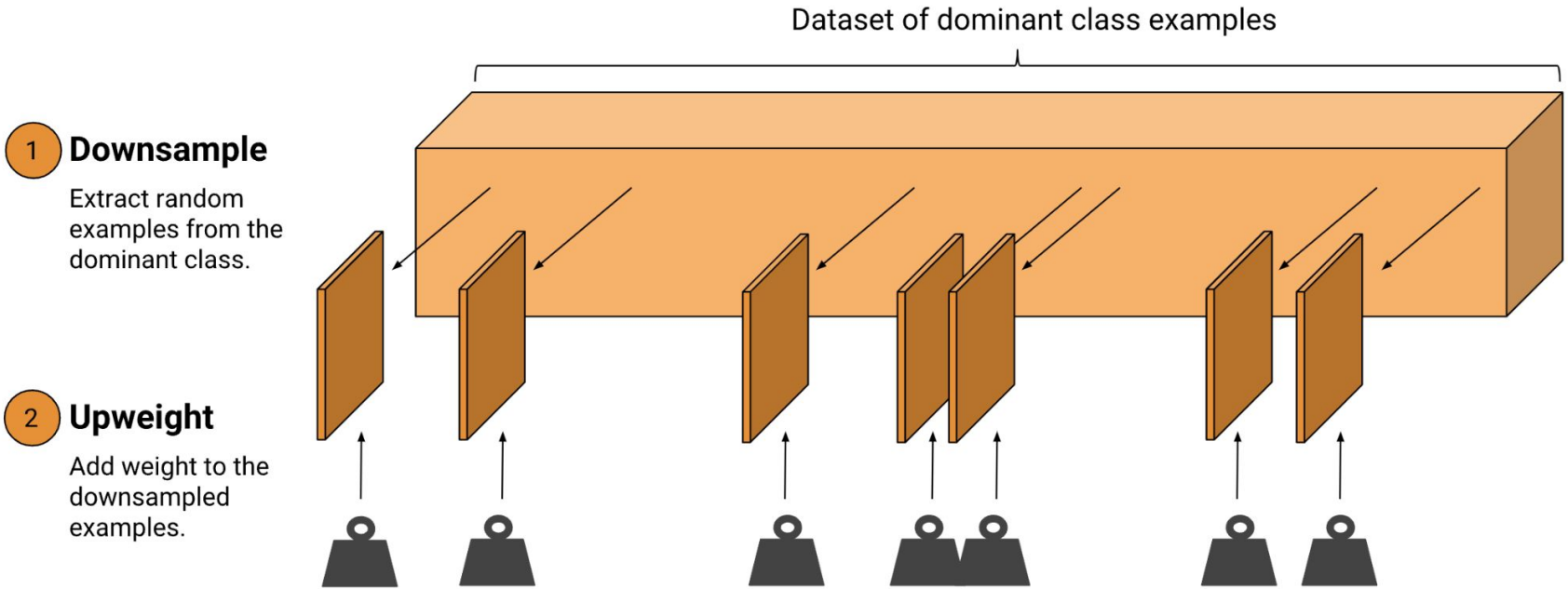
- Imbalanced Data



https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data

Understanding the data

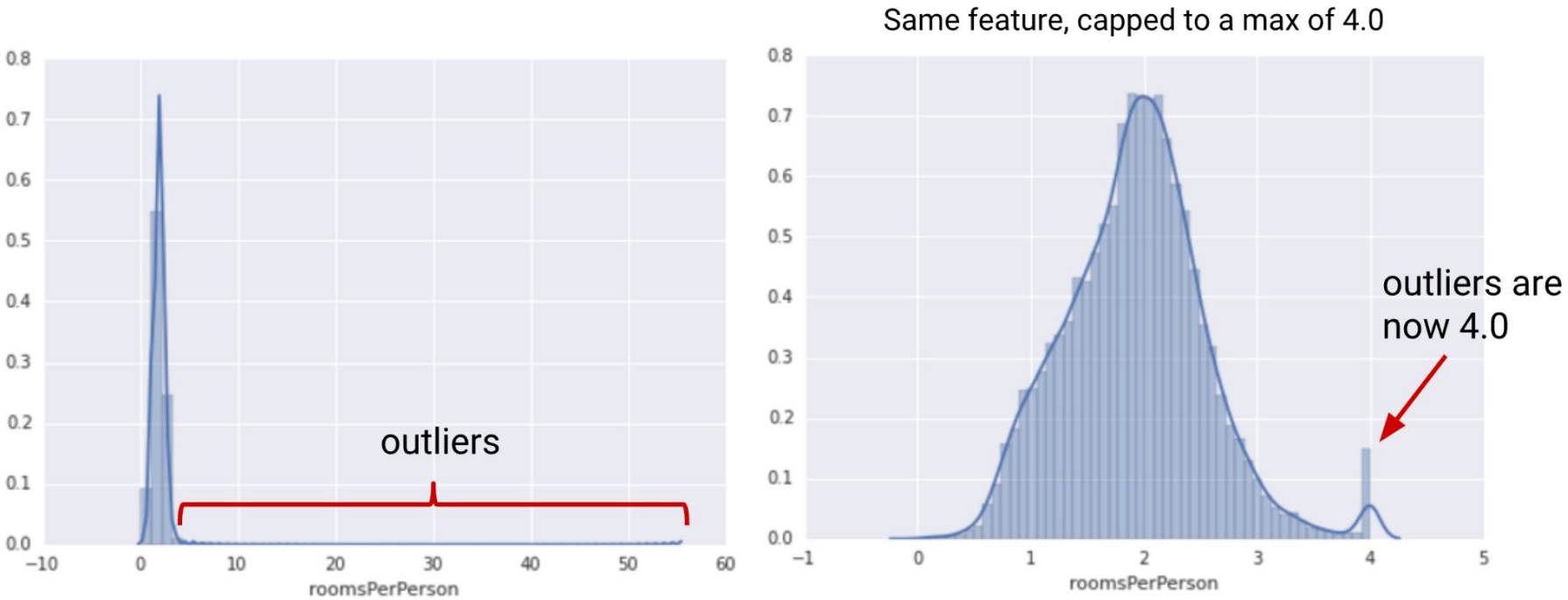
- Imbalanced Data



https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data

Understanding the data

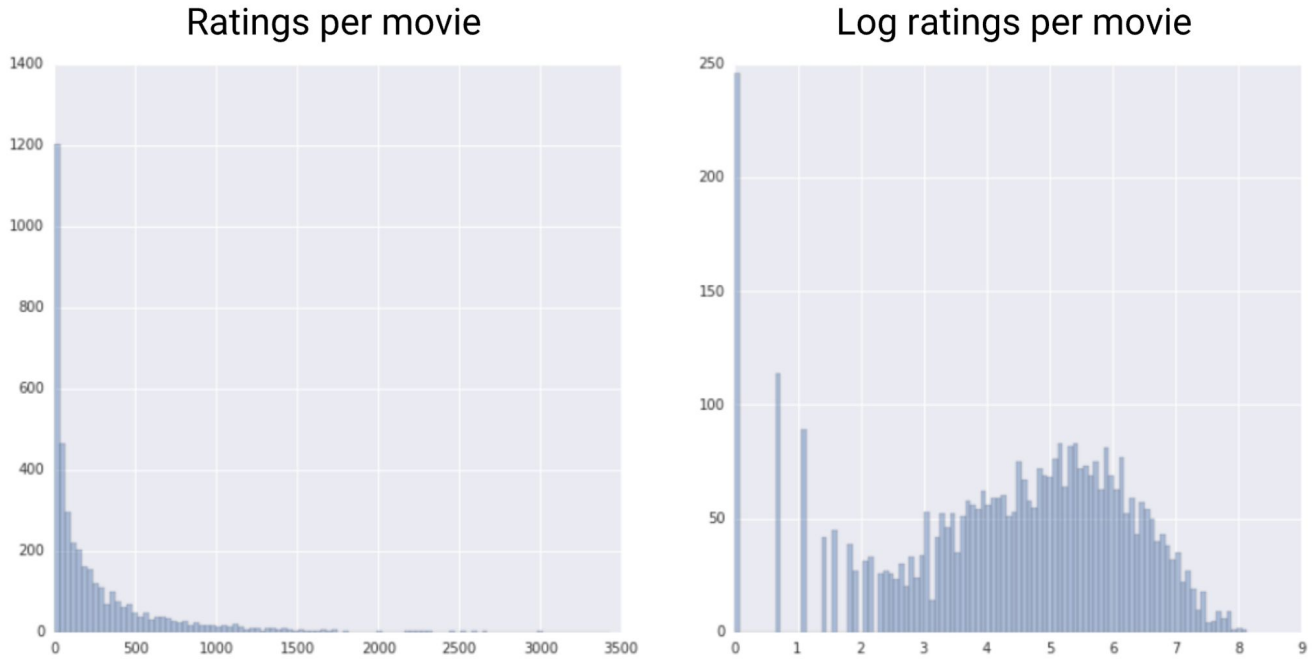
- Visualize the data through histogram



https://developers.google.com/machine-learning/data-prep/transform/normalization

Understanding the data

- Visualize the data through histogram



https://developers.google.com/machine-learning/data-prep/transform/normalization

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Platform!

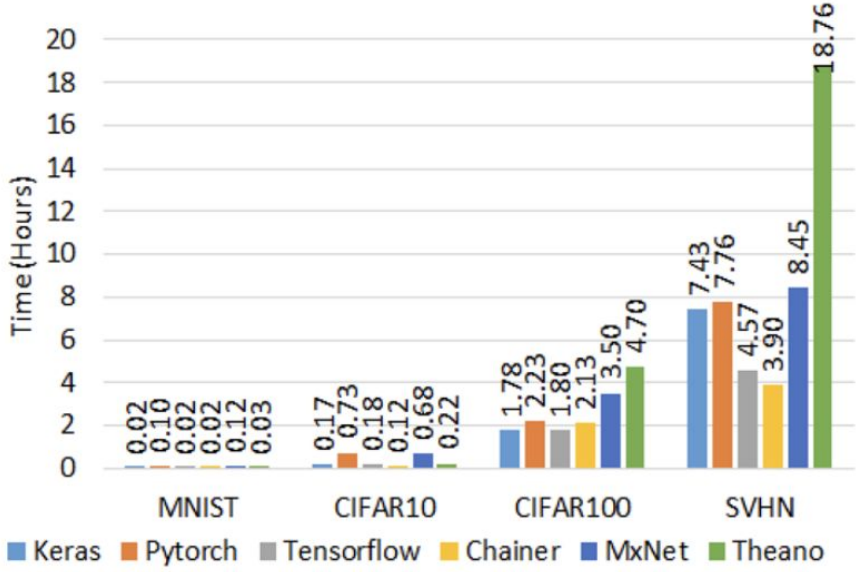
Ping-Cheng Ku

Deep Learning tools

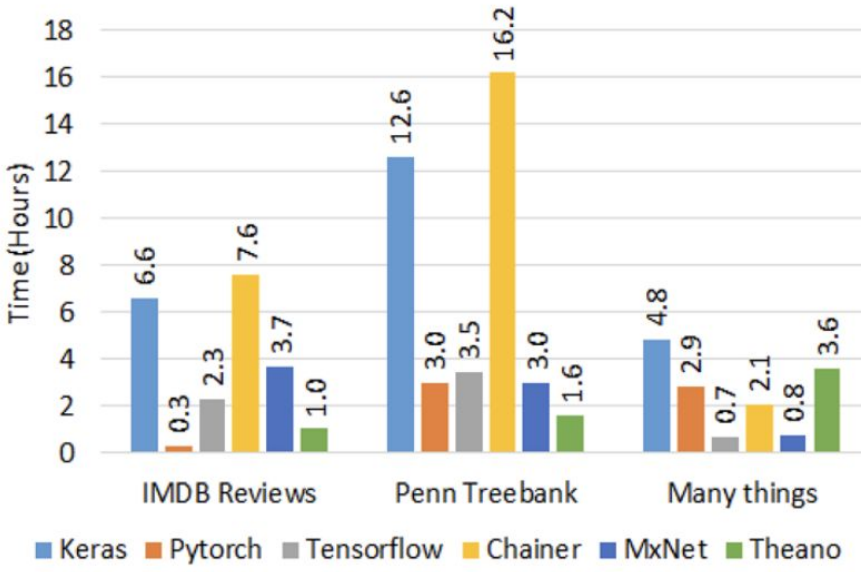
vs

Deep Learning tools

- Training time comparison (2021)



(a) CNN



(b) LSTM

https://link.springer.com/content/pdf/10.1007/s10586-021-03240-4.pdf

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Tools? Resources?

Ping-Cheng Ku

DL developer platform – WandB

Weight and Bias

- https://wandb.ai/site
- Track and visualize various aspects of model during training process in real-time
- Dataset versioning and model versioning
- Automate hyperparameter search
- Integration with both Pytorch and Tensorflow

Runs (7)

Name (7 visualized)

- jumping-breeze-7
- fancy-leaf-6
- exalted-violet-5
- rich-oath-4
- valiant-water-3
- eager-gorge-2
- resilient-shadow-1

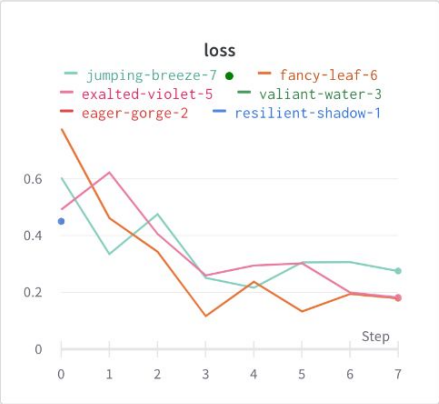
1-7 of 7

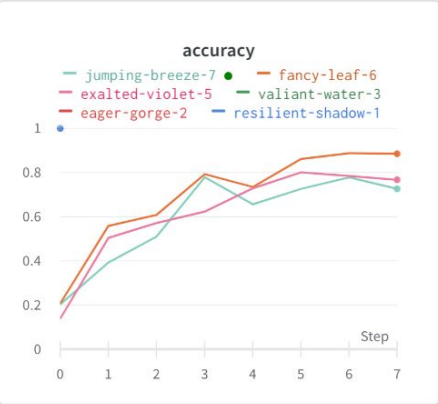
Search panels

Create report

Charts 2

Add panel





System 14

Hidden Panels 0

Add section

My Workspace

Updated 9 minutes ago

DL developer platform – Neptune AI

Neptune AI

- https://neptune.ai/
- Experiment tracking
- Dataset versioning and model versioning
- Integration with both Pytorch and Tensorflow
- Lightweight
- Better pricing for team projects
- Full comparison with WandB: https://neptune.ai/vs/wandb

DL developer platform – Tensorboard

TensorBoard

- Visualization toolkit for TensorFlow.
- Monitor model training metrics.
- Visualize computational graphs.
- View histograms of weights, biases, and activations.
- Project embeddings to lower-dimensional spaces.
- Integrates with many other ML tools.

DL resource – Papers with code

Papers with code

- https://paperswithcode.com/
- Keep track of the state-of-the-art architectures for various tasks
- Provide a wide range of dataset resources that are standardized for benchmarking (CIFAR, COCO, MNIST, Cityscapes) and corresponding research papers / codes.

DL platform – Hugging Face

Hugging Face

Hugging Face

- https://huggingface.co/
- Leading platform for NLP models.
- Transformers library: state-of-the-art architectures.
- Model hub for sharing and discovering models.
- Datasets library (mostly for NLP tasks)
- Active community contributions.

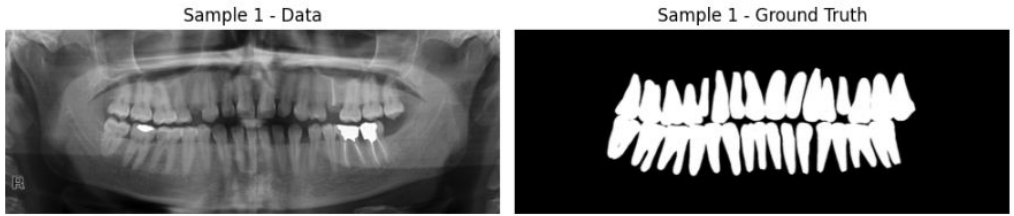
DL platform – Kaggle

Kaggle

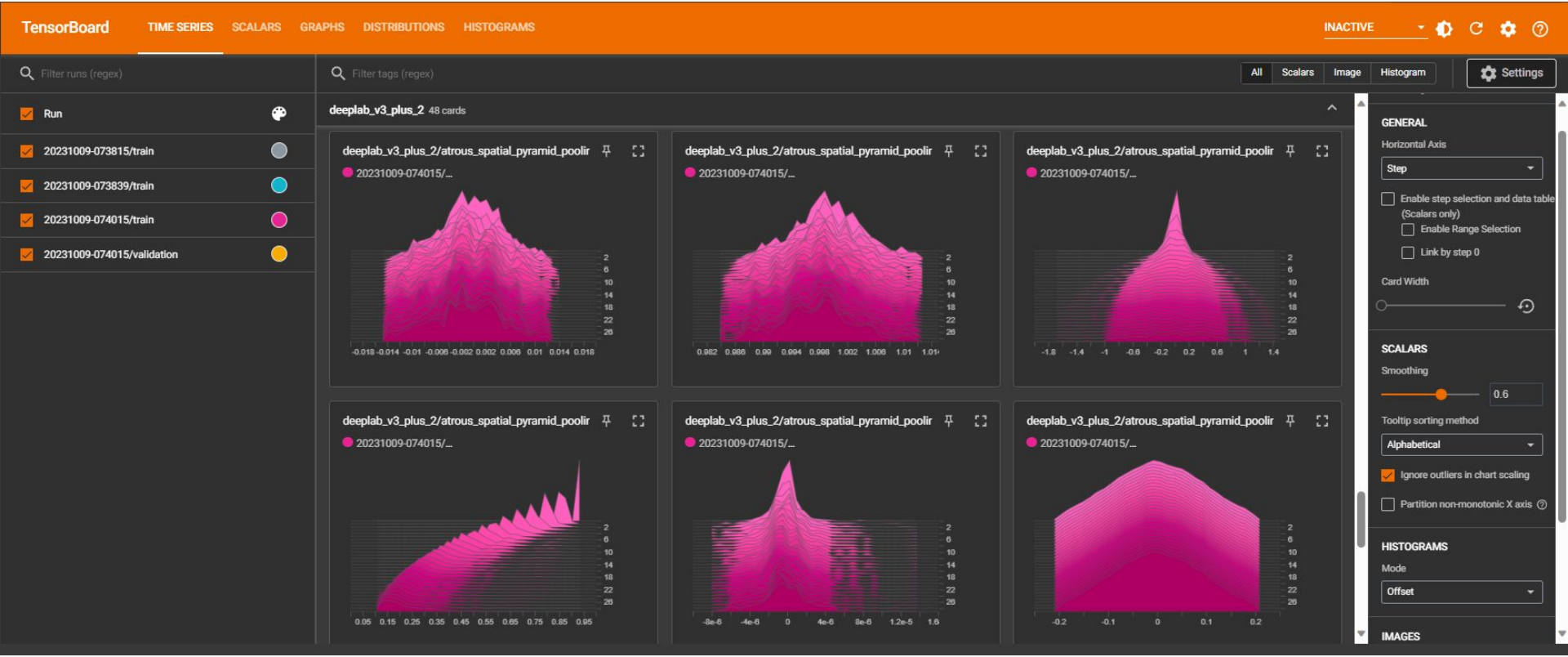
- Data science competition platform.
- Kernels/Notebooks for code sharing.
- Diverse datasets for exploration and research.
- Courses for learning data science.
- Online hosted Jupyter Notebook service

DEMO TIME!

- We will use a dataset from **HuggingFace** and use their API to process the data
- We will show how **Tensorflow** training works
- We will show how **Tensorboard** and **WandB** can be used to monitor training and compare between runs
- Colab Link: https://tinyurl.com/2jm8pfzp



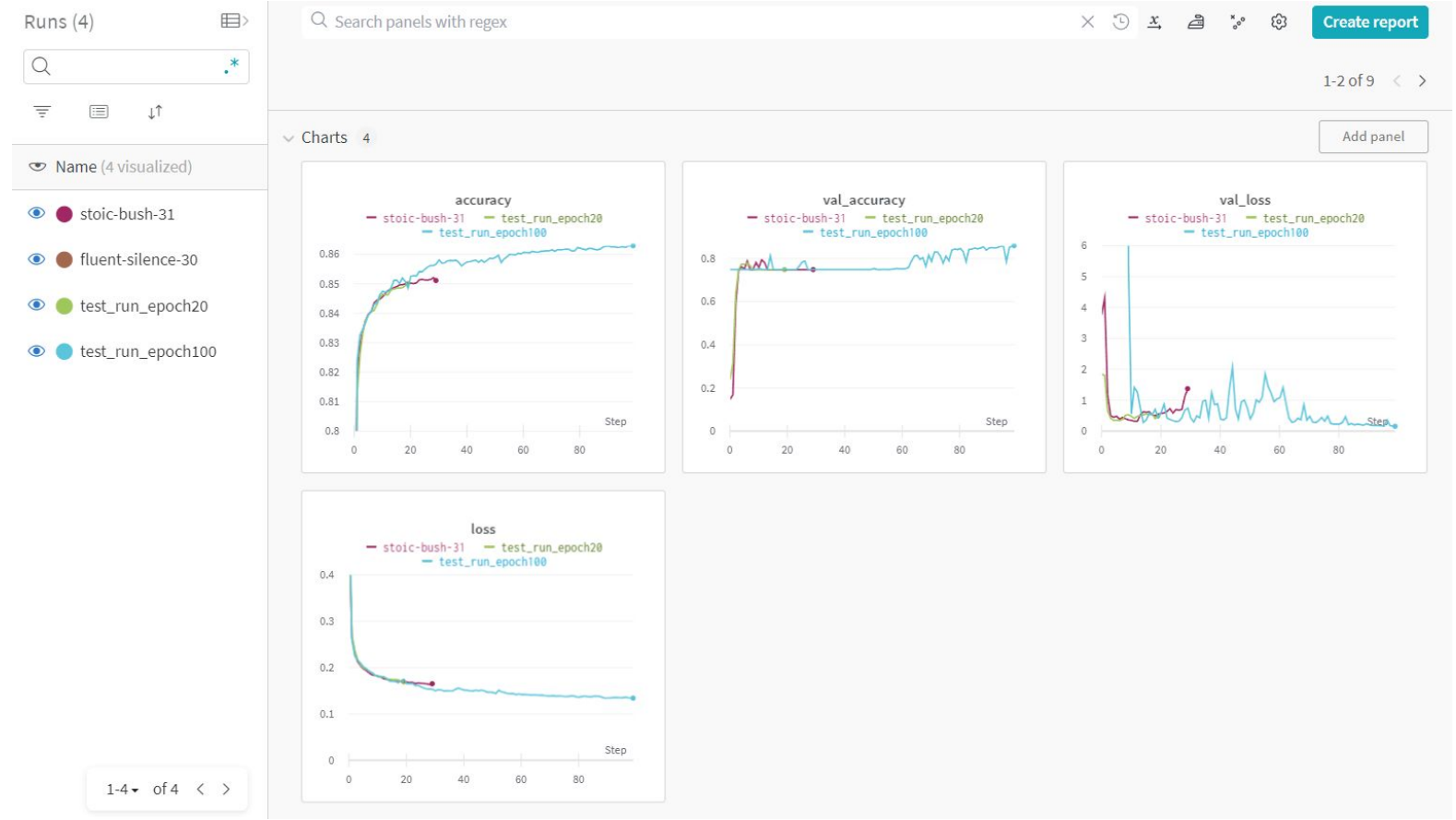
DEMO TIME!


TensorBoard
TIME SERIES
SCALARS
GRAPHS
DISTRIBUTIONS
HISTOGRAMS
INACTIVE
Filter runs (regex)
Filter tags (regex)
All
Scalars
Image
Histogram
Settings
Run
20231009-073815/train
20231009-073839/train
20231009-074015/train
20231009-074015/validation
deeplab_v3_plus_2 48 cards
deeplab_v3_plus_2/atrous_spatial_pyramid_poolir
20231009-074015/...
GENERAL
Horizontal Axis
Step
Enable step selection and data table (Scalars only)
Enable Range Selection
Link by step 0
Card Width
SCALARS
Smoothing
0.6
Tooltip sorting method
Alphabetical
Ignore outliers in chart scaling
Partition non-monotonic X axis
HISTOGRAMS
Mode
Offset
IMAGES

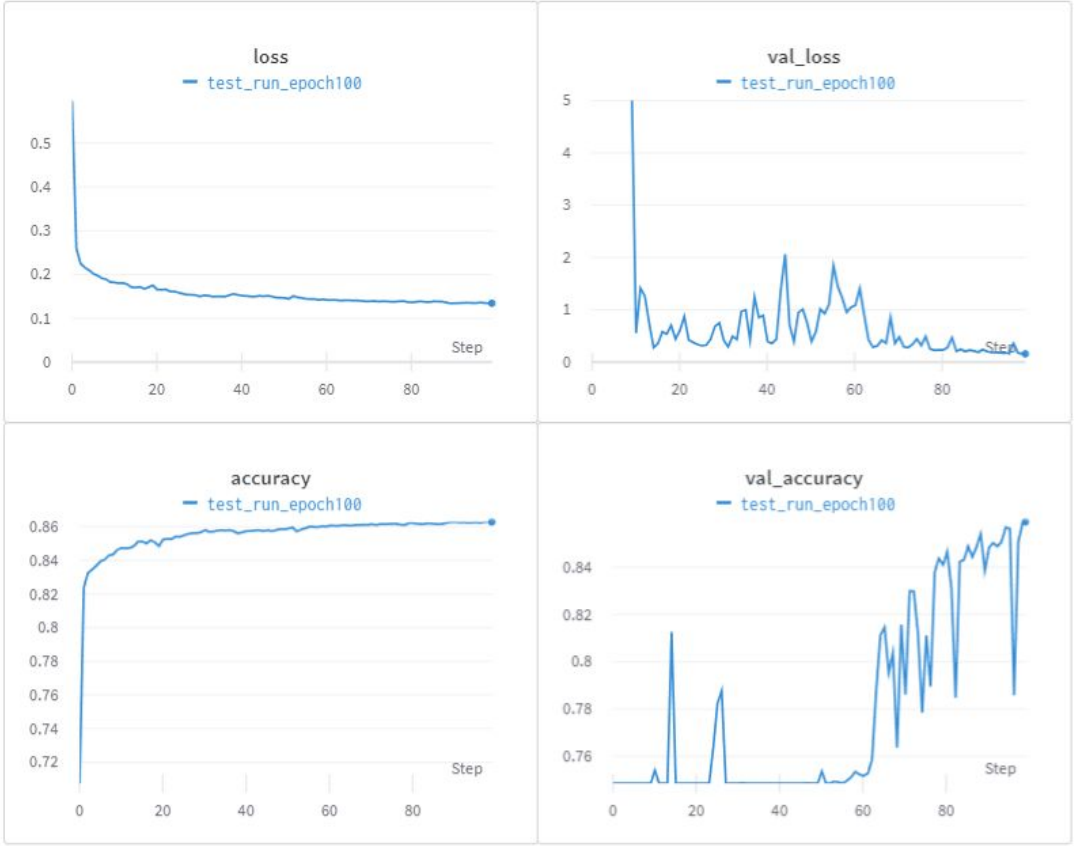
DEMO TIME!


TensorBoard
TIME SERIES
SCALARS
GRAPHS
DISTRIBUTIONS
HISTOGRAMS
INACTIVE
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
20231009-073815/train
20231009-073839/train
20231009-074015/train
20231009-074015/validation
TOGGLE ALL RUNS
logs/fit
epoch_accuracy
tag: epoch_accuracy
0.85
0.75
0.65
0.55
0
5
10
15
20
25
epoch_loss
epoch_loss
tag: epoch_loss
1.4
1
0.6
0.2
0
5
10
15
20
25
evaluation_accuracy_vs_iterations

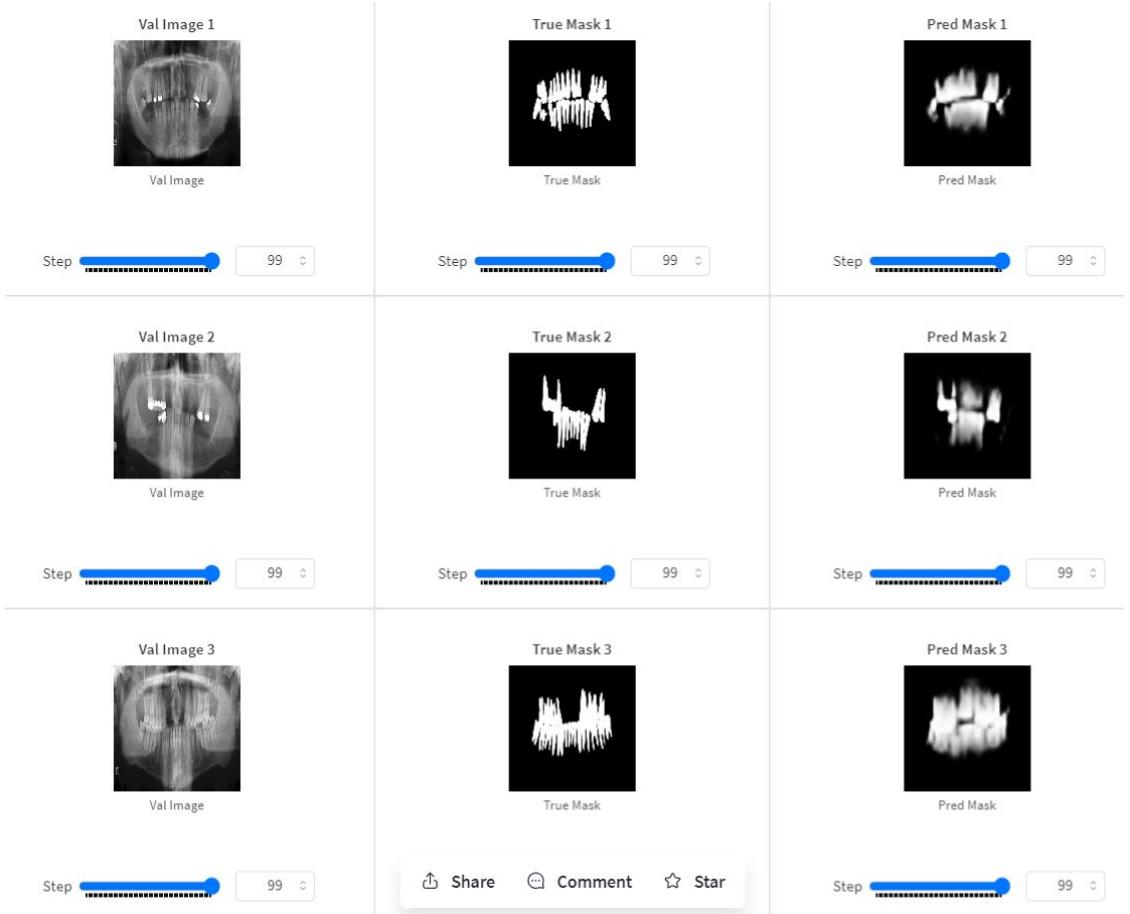
DEMO TIME!


Runs (4)
Search panels with regex
Create report
1-2 of 9
Charts 4
Add panel
Name (4 visualized)
stoic-bush-31
fluent-silence-30
test_run_epoch20
test_run_epoch100
accuracy
val_accuracy
val_loss
loss
stoic-bush-31
test_run_epoch20
test_run_epoch100
Step
1-4 of 4

DEMO TIME!


loss
test_run_epoch100
0.5
0.4
0.3
0.2
0.1
0
Step
0
20
40
60
80
val_loss
test_run_epoch100
5
4
3
2
1
0
Step
0
20
40
60
80
accuracy
test_run_epoch100
0.86
0.84
0.82
0.8
0.78
0.76
0.74
0.72
Step
0
20
40
60
80
val_accuracy
test_run_epoch100
0.84
0.82
0.8
0.78
0.76
Step
0
20
40
60
80

DEMO TIME!


Val Image 1
Val Image
Step 99
True Mask 1
True Mask
Step 99
Pred Mask 1
Pred Mask
Step 99
Val Image 2
Val Image
Step 99
True Mask 2
True Mask
Step 99
Pred Mask 2
Pred Mask
Step 99
Val Image 3
Val Image
Step 99
True Mask 3
True Mask
Share
Comment
Star
Pred Mask 3
Pred Mask
Step 99

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

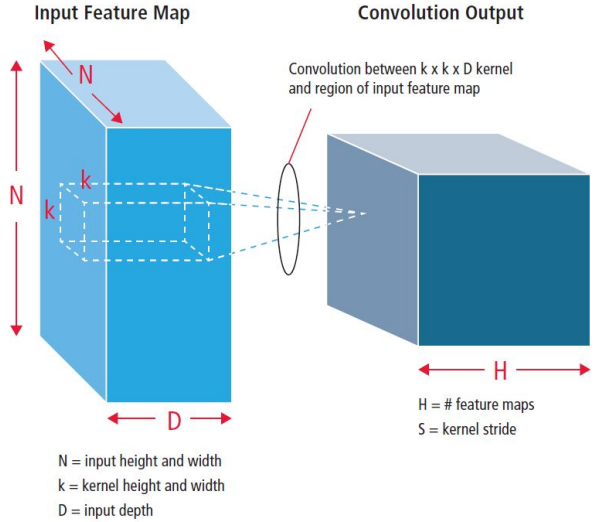
Q & A Session

Ping-Cheng Ku

Q: Calculating the number of parameters of a certain layer

- Conv2D
- Fully Connected Layers
- Batch Norm
- Activation functions
- Dropout
- Pooling

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$



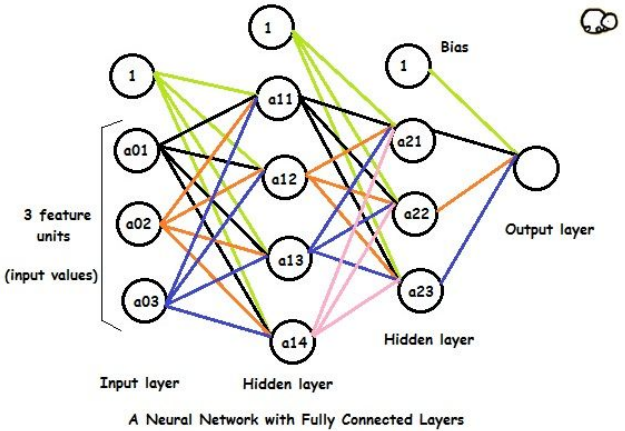


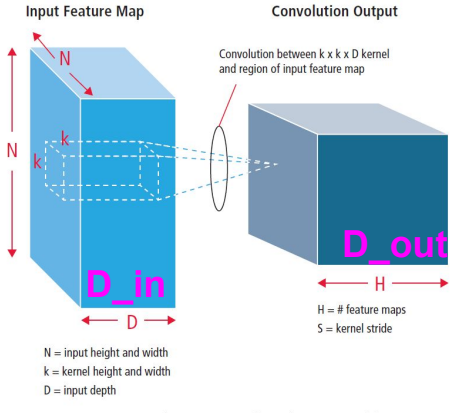
Figure 6: Pictorial representation of convolution process [9]

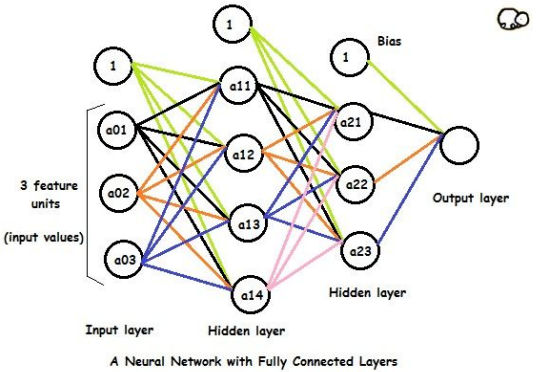
https://discuss.pytorch.org/t/how-does-conv2d-actually-work-from-inside-regarding-channel-size-of-filters/84054/2
https://iq.opengenus.org/fully-connected-layer/

Q: Calculating the number of parameters of a certain layer

kernel size

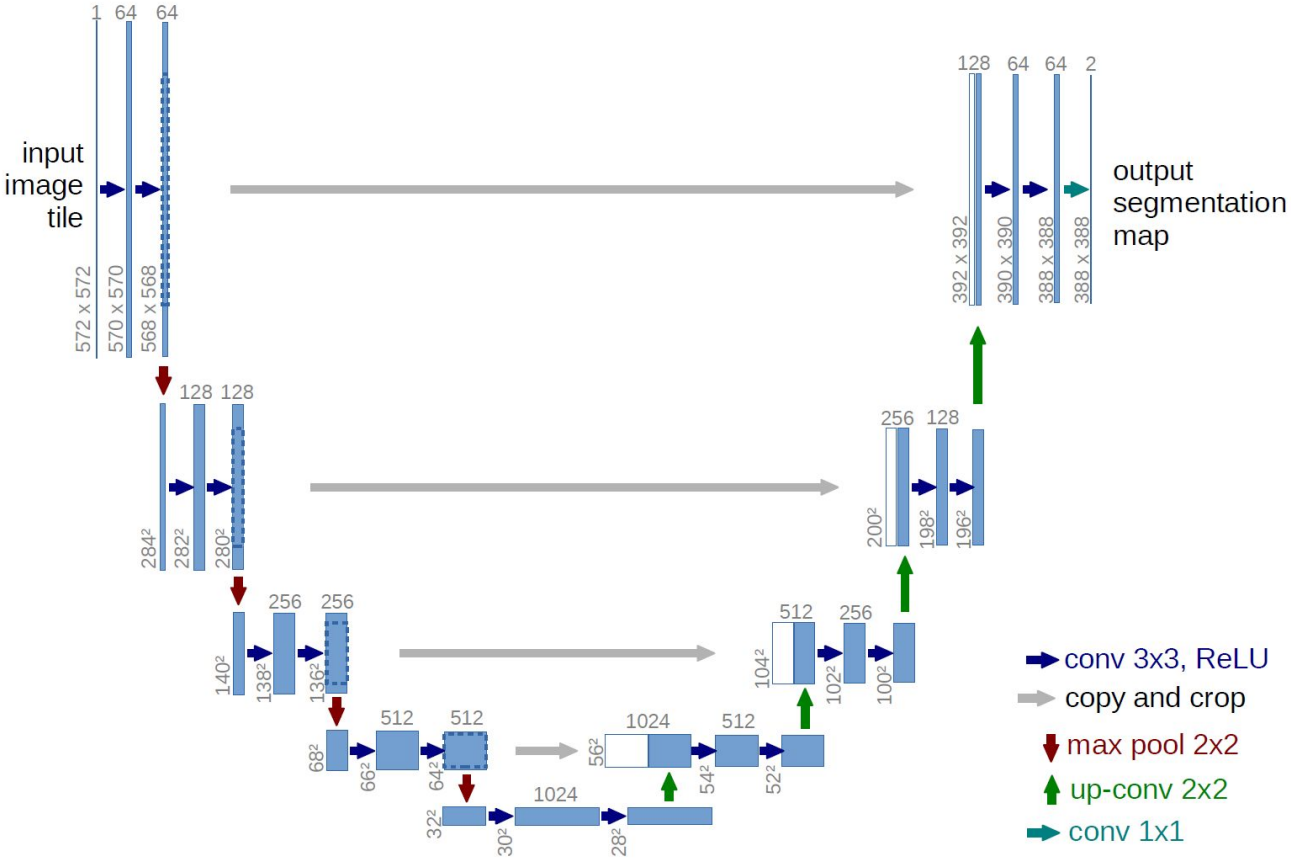
- Conv2D: (k * k * D_in + 1) * D_out
- Fully Connected Layers: (n_in + 1) * n_out
- Batch Norm: usually (2 * D_in)
- Activation functions: 0
- Dropout: 0
- Pooling: 0





https://discuss.pytorch.org/t/how-does-conv2d-actually-work-from-inside-regarding-channel-size-of-filters/84054/2
https://iq.opengenus.org/fully-connected-layer/

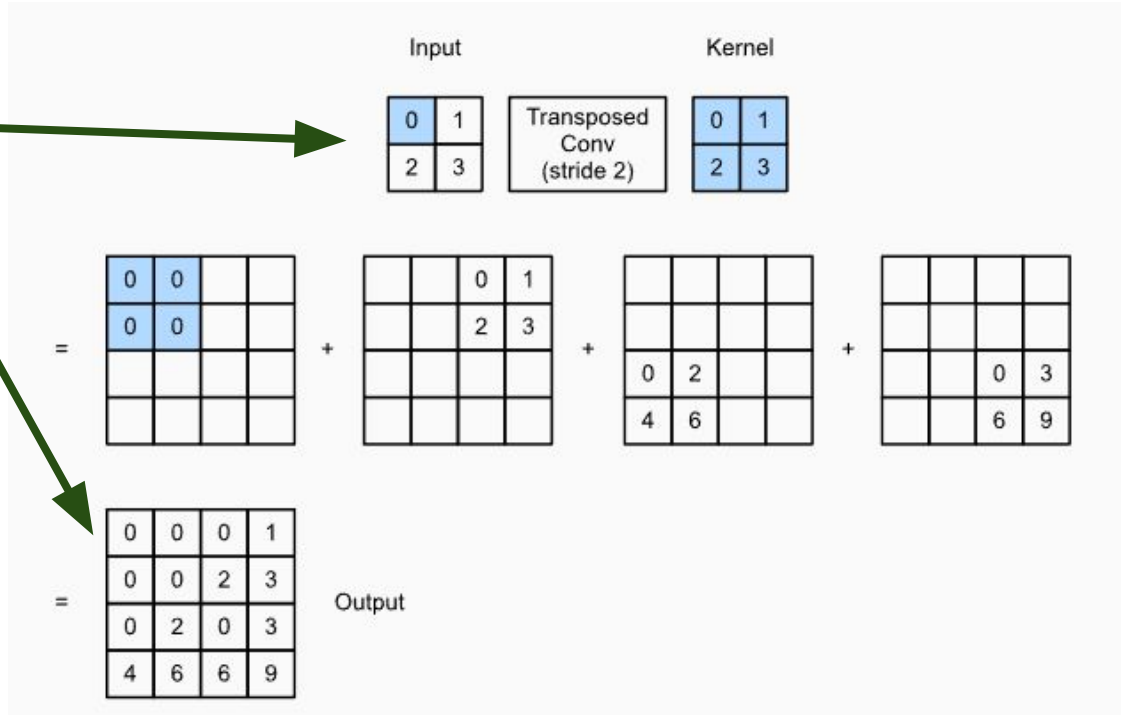
Q: Transpose Convolution/Upsampling



https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

Q: Transpose Convolution/Upsampling

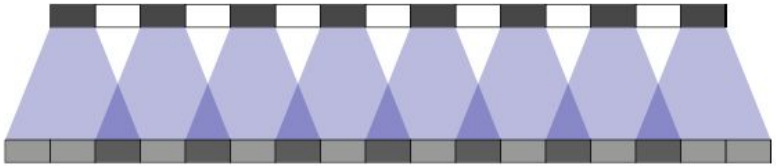
The model learns this!

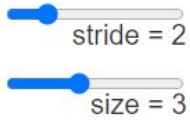


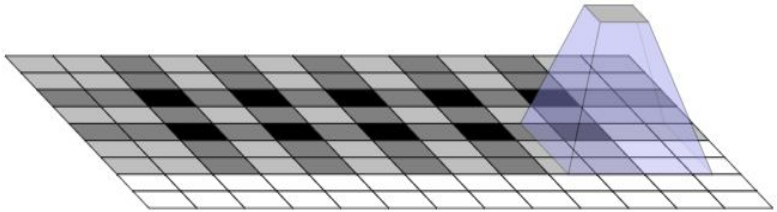
https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

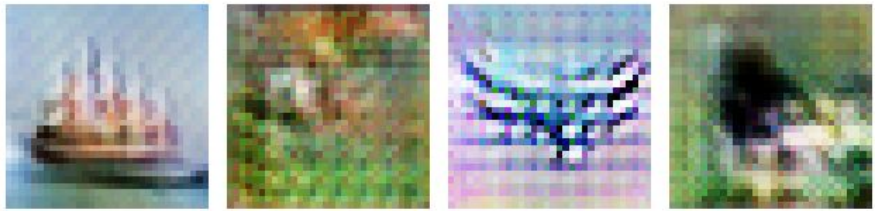
Q: Transpose Convolution/Upsampling

- Overlapping receptive field





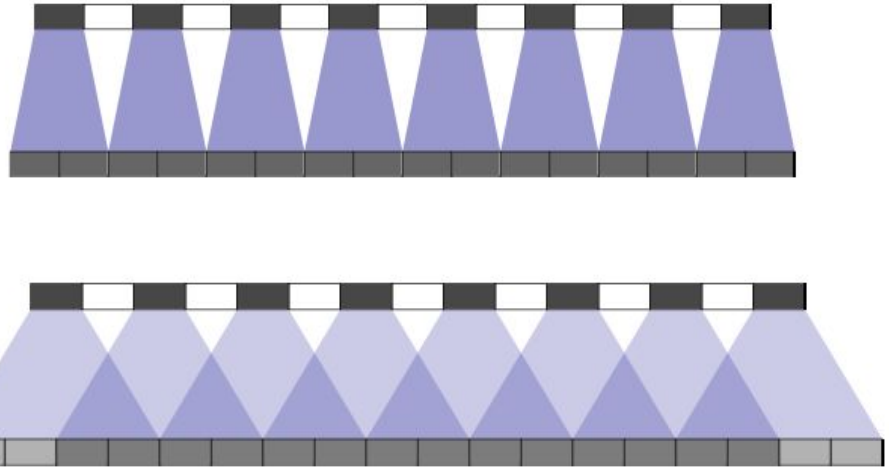




https://distill.pub/2016/deconv-checkerboard/

Q: Transpose Convolution/Upsampling

- Overlapping receptive field
  - How to mitigate: Ensure that the kernel size is divisible by the stride



https://distill.pub/2016/deconv-checkerboard/

Q: What data preprocessing strategies are recommended for MLP?

Q: Please elaborate on how we design a neural network architecture, choosing number of convolutions layers, what kinda pooling (max pooling) ...

Q: Any suggestion on possible practice material other than homework?

Q: When will the final projection session be? (when does the class end)

Week 14 - Nov 27- Midterm / Dec 13 - Final project report Due / Dec 20 - Project presentation