2021 Fast.ai Community Course

QUEENSLAND HUB

QUEENSLAND AI

Lesson 2 - Evidence and p values

Key concepts

- Validation & test datasets
- Overfitting, training, metrics/loss/error rate
- Transfer learning & fine-tuning
- CNNs & other applications
- State of Deep Learning
- P values & Null Hypothesis testing
- Begin production deployment/Bing Image Signup
- Data Block API

Training & Validation Sets

Validation & test sets are designed to simulate what is going to happen when you send your model to production.

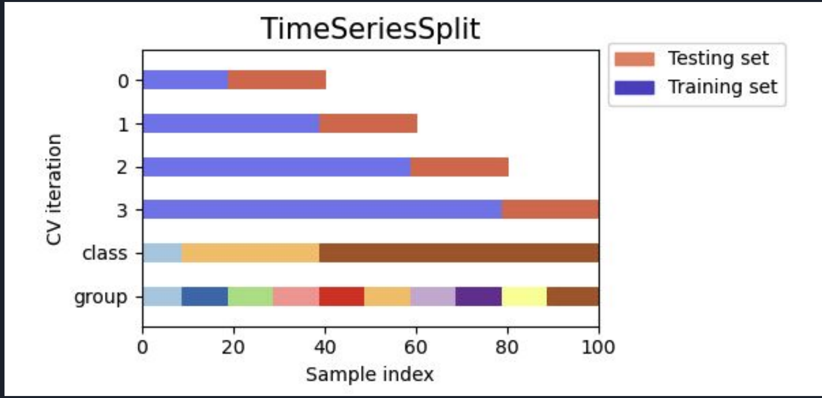
- Training set
  - Training your model
  - Metrics on training set used to understand whether your model can learn the data and if you’re overfitting
- Validation set
  - Tuning parameters & hyperparameters
  - Input features and transformations, model architectures, learning rates, data augmentation etc
  - Metrics on validation set used to tune your parameters and arrive at your final, tuned model.
- Test set
  - Produce final model accuracy metrics
  - Used <u>only once</u> to report on the accuracy of your tuned model.

Training & Validation Sets

Key assumption: That your production data is from the same distribution (IID - independently and identically distributed) as your training set. If production data is different, model performance metrics on training data will not carry over to production.

Very important: Ensure model cannot cheat due to data leakage

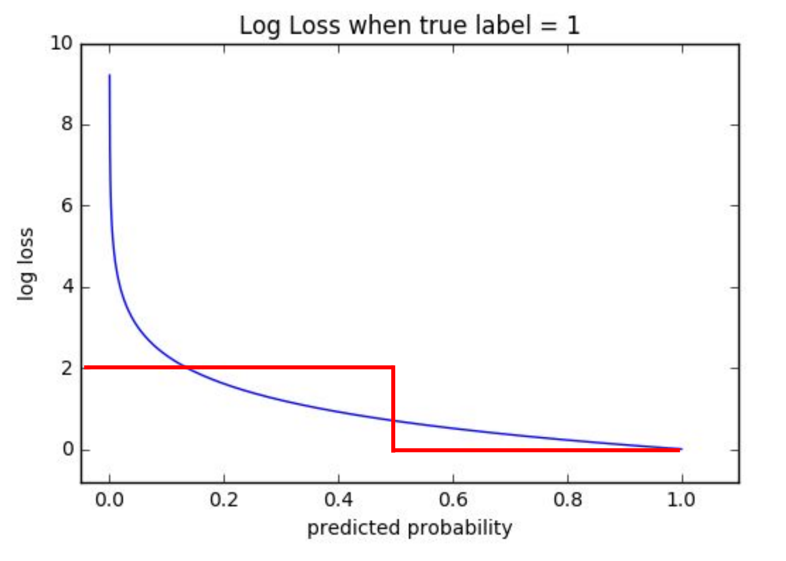
- Same patient appearing in training & test set: Split into sets based on by patient ID, then assign all samples for each patient to the correct group
- Time Series: Walk forward chronologically, getting new validation metrics on new chunks of data. Then take the average of performance across all validation sets.
- [Link: visualisation of different approaches]



Training Loss vs Accuracy Metrics

- **Loss Function:** A smooth function that allows will show minute improvements in accuracy as the model improves. Example is binary cross-entropy loss (BCE), AKA Log Loss
- **Evaluation Metric:** A measure that communicates the accuracy/performance of the model (e.g. accuracy, precision, recall, error rate).

True Class	Pred Class	Accuracy	Log Loss
1	0.3	0	1.204
1	0.45	0	0.799
1	0.6	1	0.511
1	0.75	1	0.288
1	0.9	1	0.105



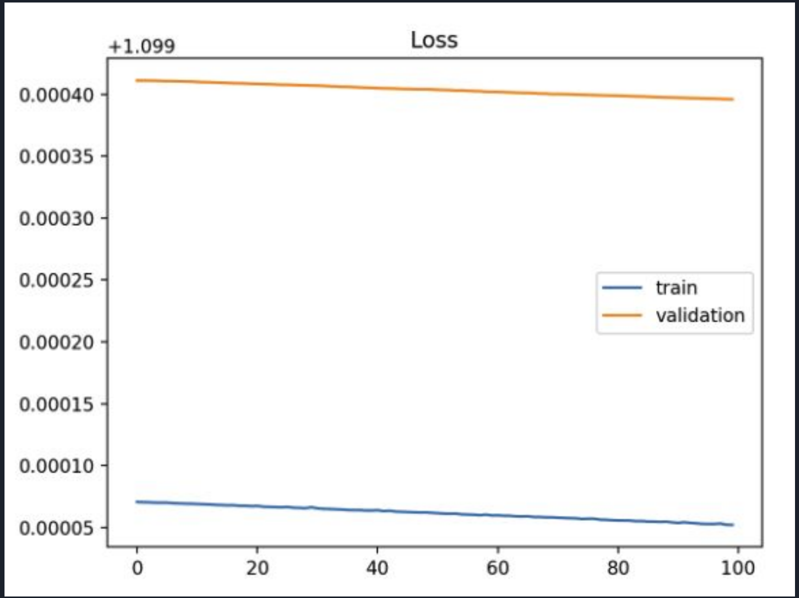
Loss Metrics & Overfitting

Rules of thumb:

- **Keep training loss and validation loss close together**
  - If not, model may not be powerful enough, or dataset large enough to capture enough variance
- **Training loss and validation loss should decrease smoothly**
  - If unsmooth, dataset may be too small, not shuffled correctly
- **Training and validation loss should gradually flatten out**
  - If they have not flattened, the model requires more training
  - Note: Even if metrics appear to have flattened out, the model could still become more accurate with a smaller learning rate or further training. Don't stop training until performance on the validation set consistently decreases.

Tips for training deep nets: https://karpathy.github.io/2019/04/25/recipe/

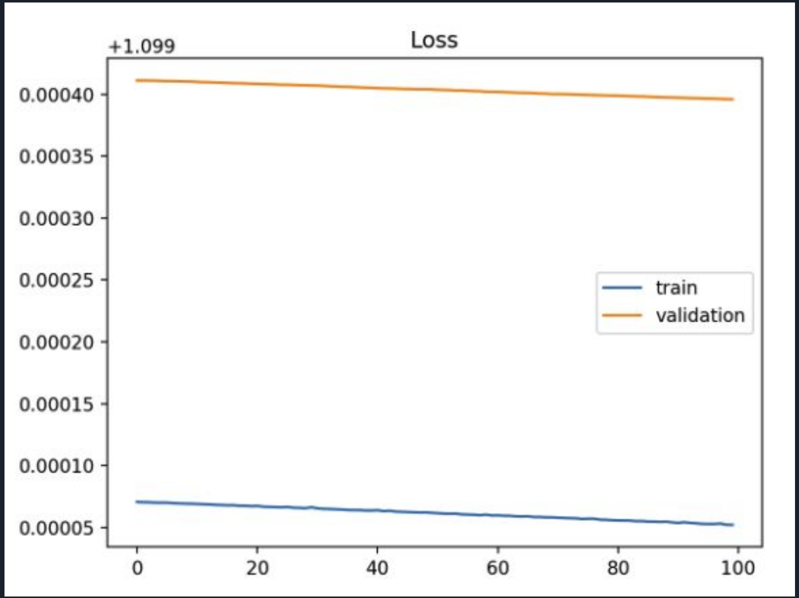
Loss Metrics & Overfitting



Diagnosis: ??

Source: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance

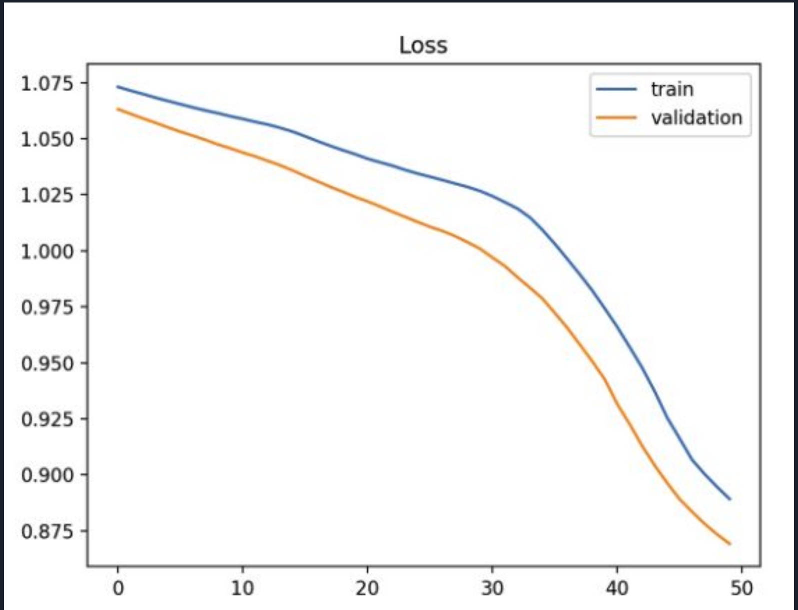
Loss Metrics & Overfitting



Diagnosis: An Underfit Model That Does Not Have Sufficient Capacity

Source: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance

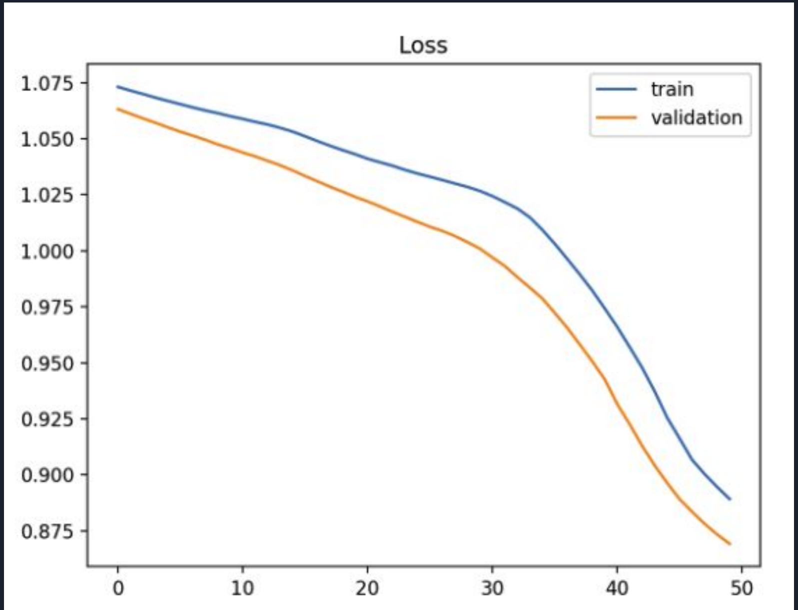
Loss Metrics & Overfitting



Diagnosis: ??

Source: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance

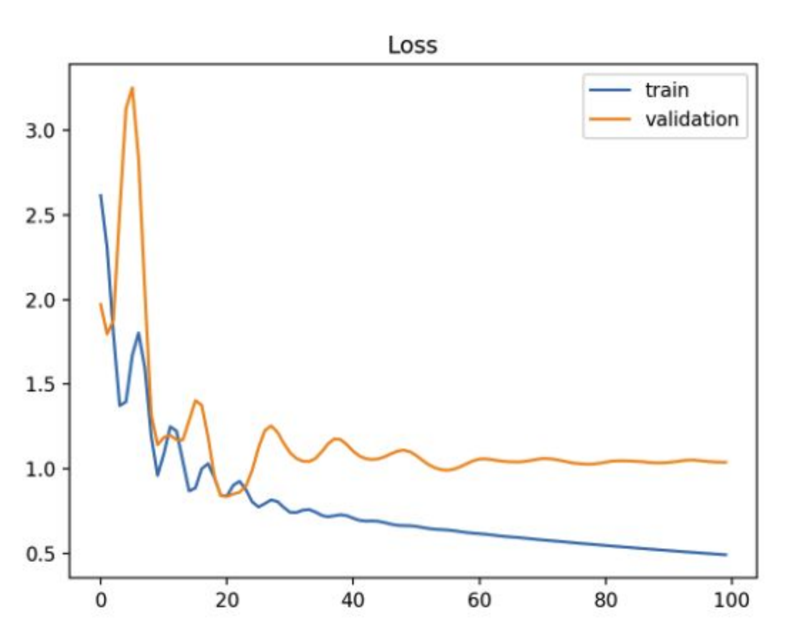
Loss Metrics & Overfitting



Diagnosis: An Underfit Model That Requires Further Training

Source: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance

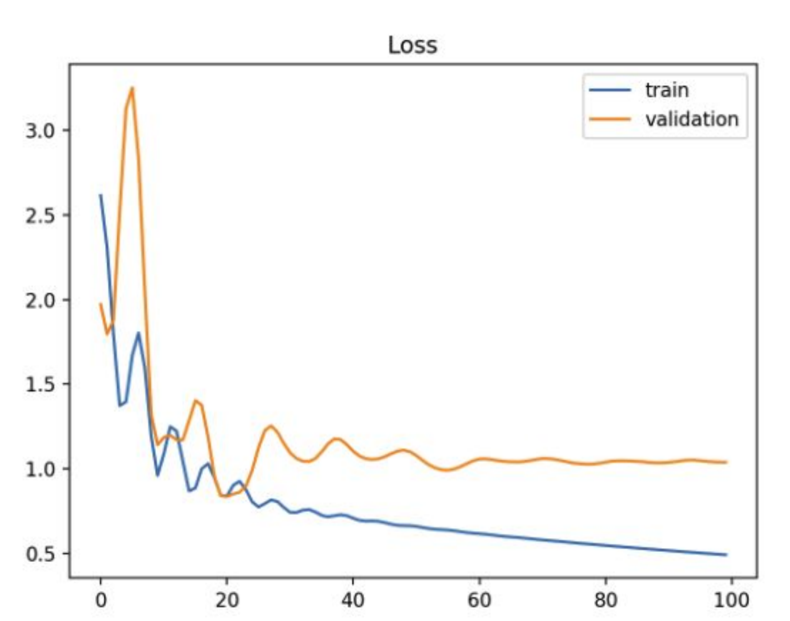
Loss Metrics & Overfitting



Diagnosis: ??

Source: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance

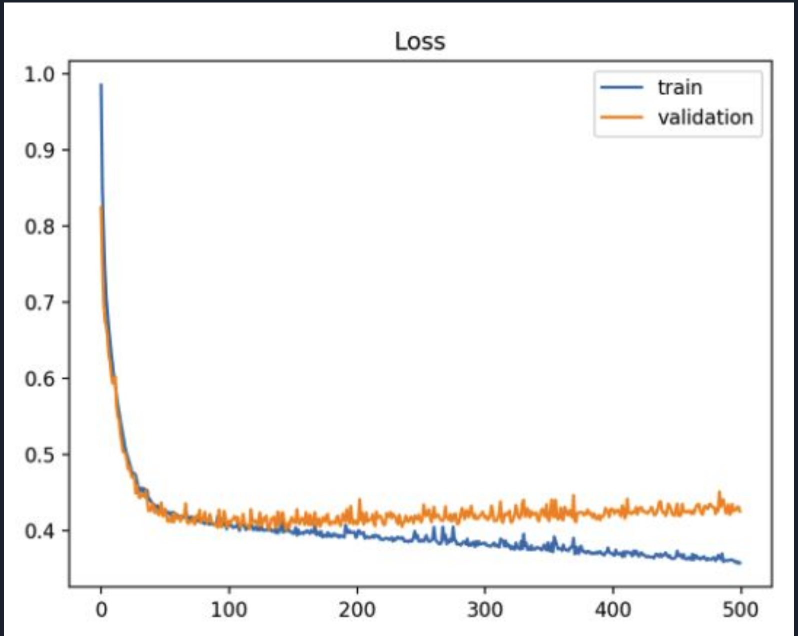
Loss Metrics & Overfitting



Diagnosis: A Training Dataset That May Be too Small Relative to the Validation Dataset

Source: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance

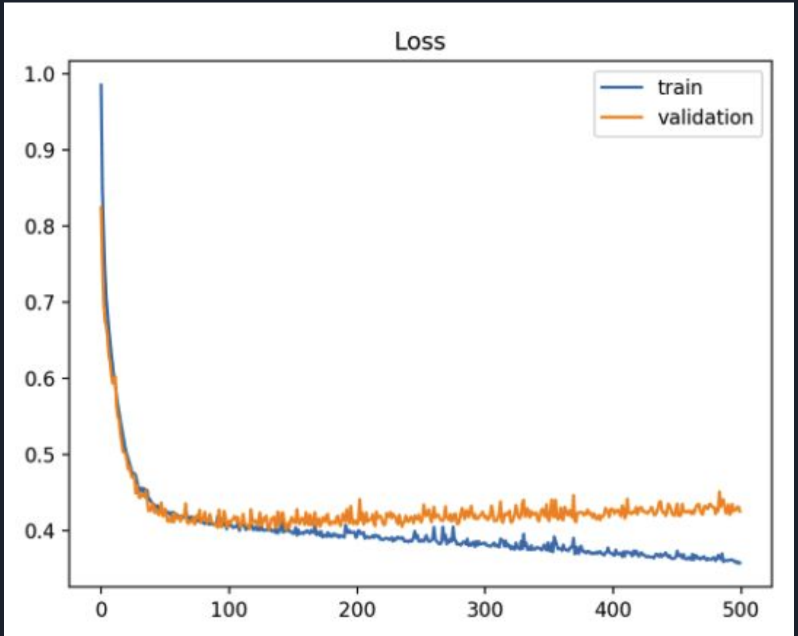
Loss Metrics & Overfitting



Diagnosis: ??

Source: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance

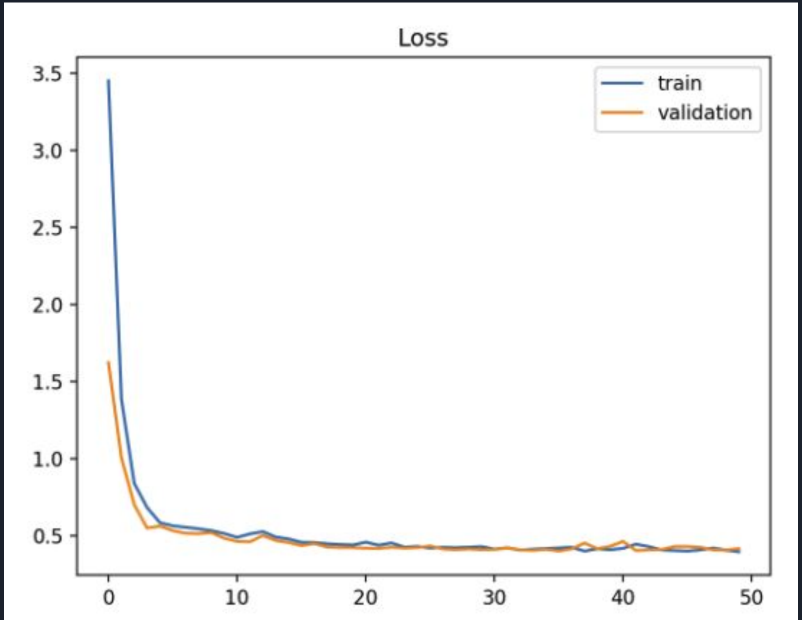
Loss Metrics & Overfitting



Diagnosis: An overfitting model

Source: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance

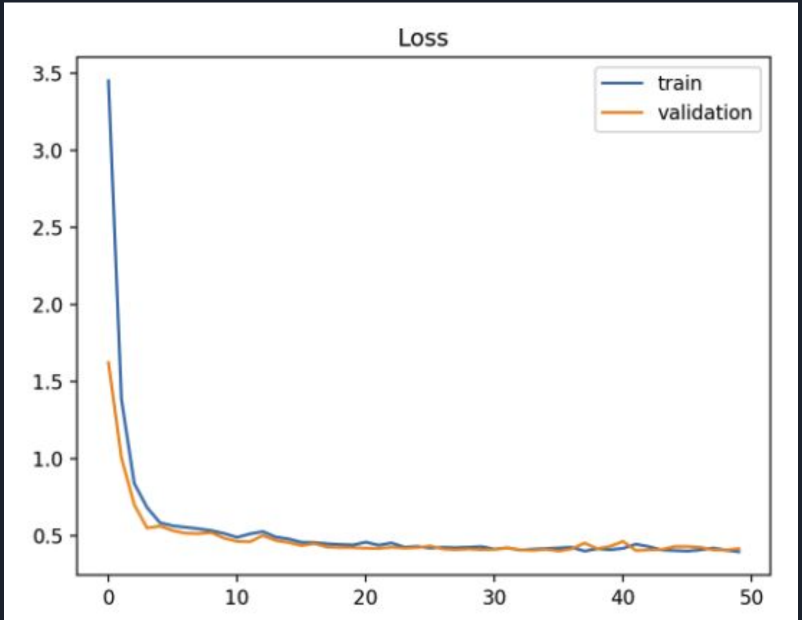
Loss Metrics & Overfitting



Diagnosis: ??

Source: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance

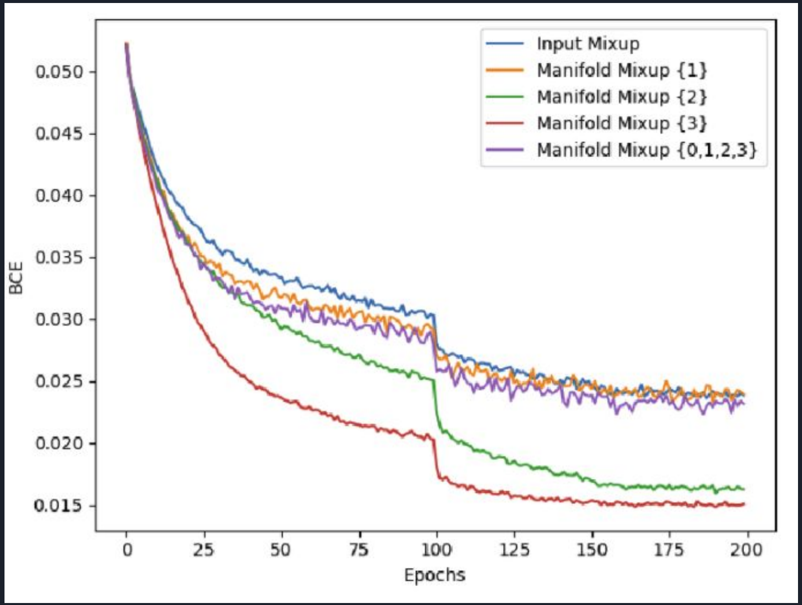
Loss Metrics & Overfitting



Diagnosis: A Good Fit (this is what we are looking for)

Source: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance

Loss Metrics & Overfitting



Extra Notes: An example of how dynamically changing learning rates or model parameters can lead to steps in loss metrics

Source: https://www.researchgate.net/publication/325778354_Manifold_Mixup_Encouraging_Meaningful_On-Manifold_Interpolation_as_a_Regularizer

Transfer Learning

Standard practice across most domains (CV & NLP in particular)

Use another, larger dataset to learn representation of data, then reuse that representation on our dataset.

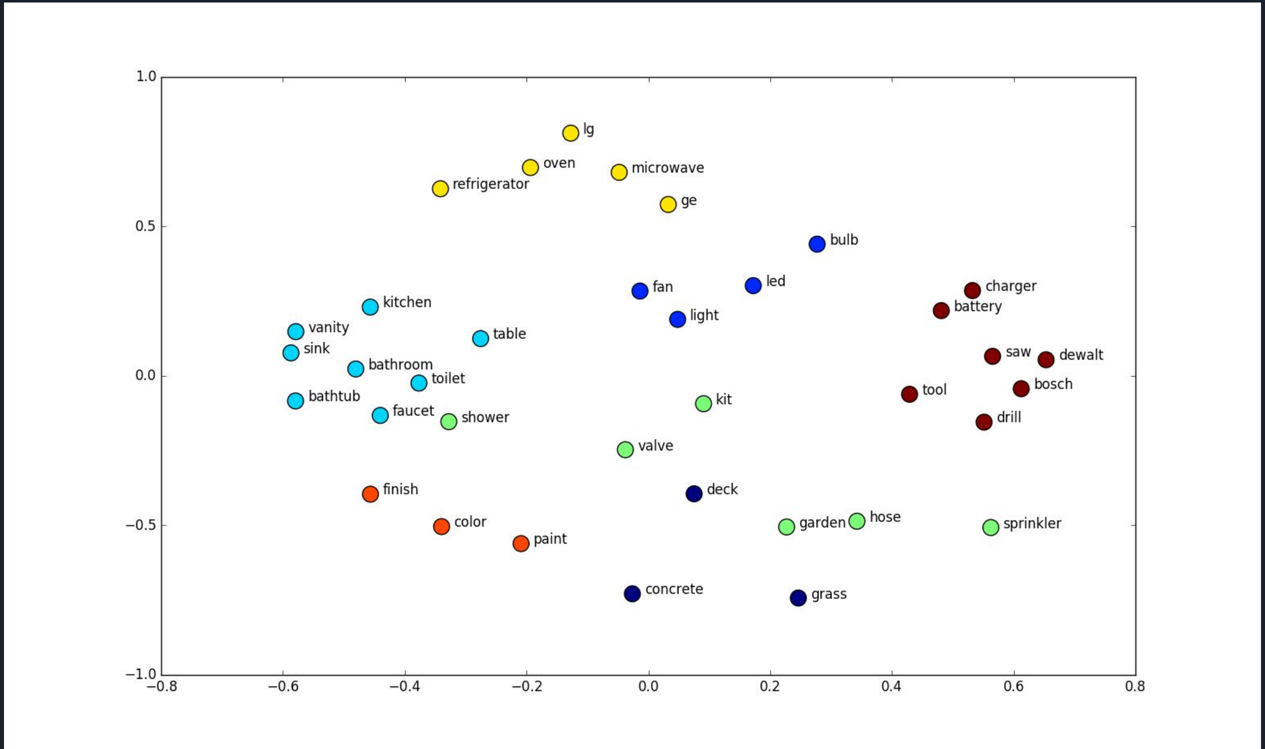
Tip: Try to use a dataset that has similar style, context and classes as your own. This ensures the layers will require only small fine tuning

- If your problem is aerial imagery, try to use an aerial imagery dataset.
- If your problem is Medical CT scans, find a CT scan dataset.

Interesting links:

- https://jalammar.github.io/illustrated-word2vec
- https://distill.pub/2017/feature-visualization
- https://distill.pub/2021/multimodal-neurons (advanced)

Transfer Learning Representation



Model Architectures

A model architecture determines how information will be processed. You are encoding your domain knowledge about the input data into the model.

Certain architectures are able to better extract semantic meaning from the input data. The breakthrough of CNNs was to use convolutional filters to process adjacent areas of the image together.

To find appropriate architectures for your dataset, visit paperswithcode.com and look at the top papers for your problem. Most libraries (including fast.ai) will have the best practice models implemented.

Newer architectures typically give small accuracy improvements, or larger speed improvements

Managing Risk in Production

Machine Learning failures can happen silently but have tremendous impact

- Models can be making high risk, high impact decisions
- Models are sensitive. Data distribution changes over time can leading to a model that has never seen the data before making predictions.

Ways to manage risk

- Always have regular model retraining and effective monitoring (of both the data and the model)
- Use human-in-the-loop approach: Have model make a prediction, then a human review it
- Use model explainability techniques: Give an explanation of which features contributed most to the decision, or show similar training examples in the training data as a reference.
- Give measures of model uncertainty along with the prediction.
    - In simple terms, model uncertainty is a measure of how well calibrated the model is to the example it has made a prediction on. If the model has not seen any or only a few examples of that input data, it should return a high model uncertainty metric.
- Use confidence or uncertainty thresholding: Automatically process all predictions with high confidence or low uncertainty score, and send all the remaining to a human.
    - E.g. automatically process all predictions < 0.1 or > 0.9, and have humans review everything in between.

Data Loaders

DataLoaders provide an optimised way to load, organise, transform and feed data to the GPU. Data Loaders are a key component of all neural network libraries (Pytorch, Tensorflow etc)

Fast.ai DataBlock class gives a simple interface to build our training and test data loaders.

- Loads the data (get_items)
- Takes the data and splits it into x and y (get_x, get_y)
- Organises the input data into required blocks for the neural network, organised by data types and target types (blocks)
- Transforms the data (item_tfms)
- Splits the data into separate training and validation sets (splitter)

```
bears = DataBlock(
    blocks=(ImageBlock, CategoryBlock),
    get_items=get_image_files,
    splitter=RandomSplitter(valid_pct=0.3, seed=42),
    get_y=parent_label,
    item_tfms=Resize(128))
```

Break for Questionnaire

Group Projects

This week we will start organising project groups.

- Teams can be formed of both in-person and online participants
- You should organise together based on timezone (e.g. Asia/Americas/Europe) and data domain
- Teams will be confirmed next week, June 15th.
- Instructions will go up tomorrow in the #group-projects discord channel

- If you would like to use your own dataset, the data must:
  - Be available for public use, or be given with written permission
  - Not require any approvals or ethics courses (e.g. research datasets)
  - Be applicable to the course (problem is constrained for 6 weeks of work, use the fast.ai library)
- If it meets that criteria, you can seek approval by sending the required information (provided in the #group-projects discord channel) to a mentor.

Group Projects

A list of example datasets will be available in a couple hours:

Think outside the box, trying to do more than just build a standalone model. Try to produce something that can be interacted with. For example:

- Face emotion recognition with webcam
- Capture speech with audio, convert with voice to text service, classify with fast.ai
- Predict someone’s personality type in the browser based on questions
- Sign language recognition with webcam