CS/ENGR M148 L17:
NN Interpretability

Sandra Batista

Administrative News

This week in discussion section:

Neural Network Project Check-in. Applying NN to project data for classification or prediction. Reporting metrics. This can be what you include in final project report.

There will also be review for final exam.

PS4 due today and quiz Thursday.

Final project report guidelines posted.

Final Exam Practice Questions posted.

Join our slido for the week…

https://app.sli.do/event/nCV57u4mC7eUMit9euSBr2

Today's Learning Objectives

Students will be able to:

- Review Exercise: NN Training and Learning Rate
- Review: Regularization for NN
- Ensembles and Dropout
- Interpretability for NN

NN Training Review Exercise

You are training a neural network and you are trying to figure out what to use for the **learning rate** hyperparameter to gradient descent.

What do you do using 100 training iterations (without just using PyTorch)?

(During discussion this week you will review Learning Rate Schedulers in PyTorch)

Aurelien Geron. 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd. ed.). O'Reilly Media, Inc

NN Training Review Exercise

How to figure out hyperparameter: learning rate

Aurelien Geron. 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd. ed.). O'Reilly Media, Inc

NN Training Review Exercise

How to figure out hyperparameter: learning rate

Aurelien Geron. 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd. ed.). O'Reilly Media, Inc

Today's Learning Objectives

Students will be able to:

-  Review Exercise: NN Training and Learning Rate
- Review: Regularization for NN
- Ensembles and Dropout
- Interpretability for NN

What is regularization?

Regularization *is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error.*

Regularization of NN

1. Norm penalties
2. Early Stopping
3. Data Augmentation
4. Dropout

Norm Penalties

We used to optimize:

$$W^{(i+1)} = W^{(i)} - \lambda \frac{\partial L}{\partial W}$$

Change to:

$$L_R(W; X, y) = L(W; X, y) + \frac{1}{2}\alpha \parallel W \parallel_2^2$$

$$W^{(i+1)} = W^{(i)} - \lambda \frac{\partial L}{\partial W} - \lambda\alpha\, W^{(i)}$$

Weights decay in proportion to size

Biases not penalized

L_2 regularization:

- Weights decay

$$\Omega(W) = \frac{1}{2} \parallel W \parallel_2^2$$

L_1 regularization:

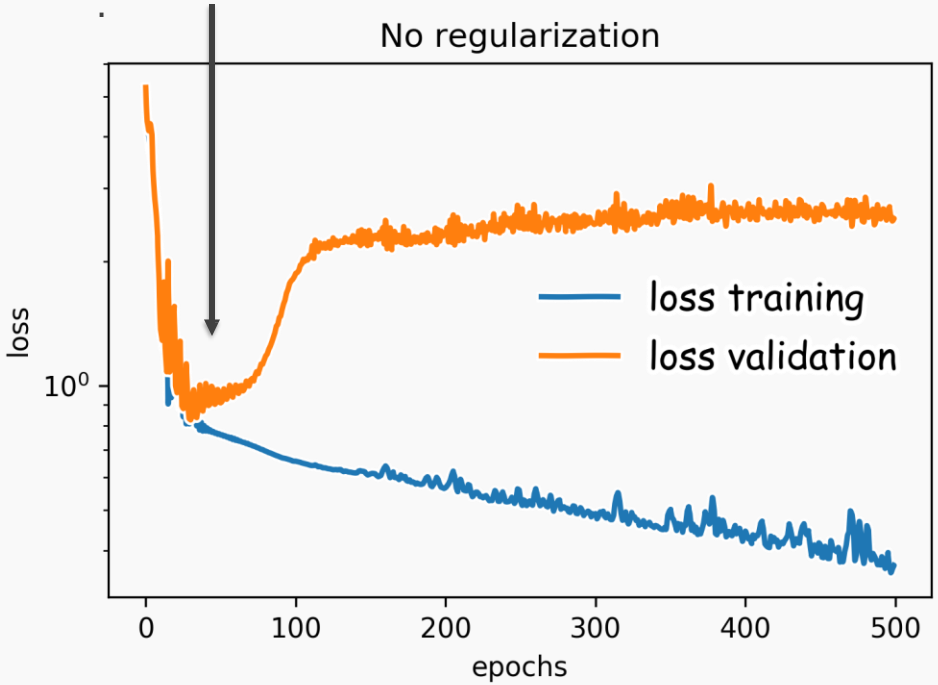
- encourages sparsity

$$\Omega(W) = \frac{1}{2} \parallel W \parallel_1$$

[CS M148 Winter 2024]

Early Stopping

Early stopping: terminate while validation set performance is better. Stop training when validation error reaches a minimum
Hinton calls this a “beautiful free lunch”



Patience is defined as the number of epochs to wait before early stop if no progress on the validation set.

The patience is often set somewhere between 10 and 100 (10 or 20 is more common), but it really depends on the dataset and network.

[CS M148 Winter 2024]

Aurelien Geron. 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2nd. ed.). O'Reilly Media, Inc

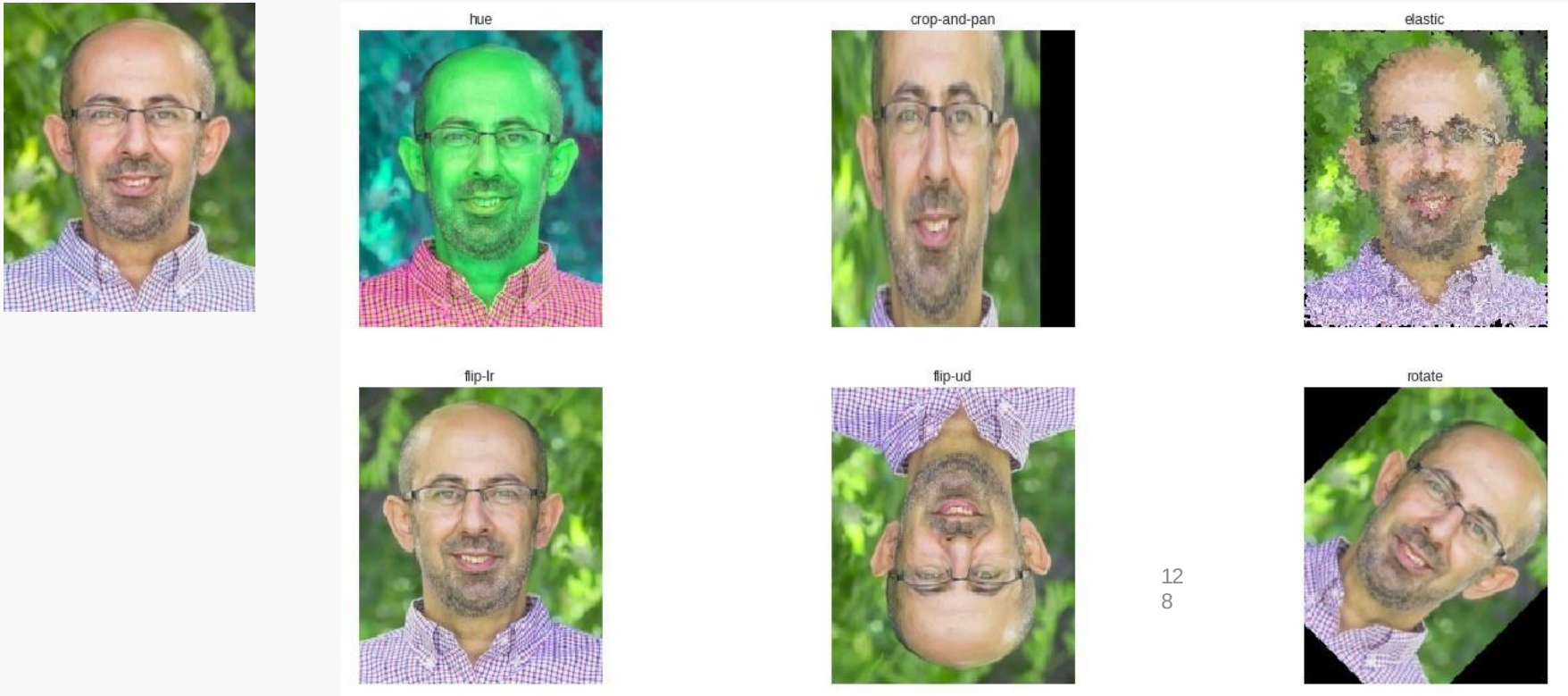
Early Stopping

Let n be the number of steps between evaluations.
Let p be the "patience," the number of times to observe worsening validation set error before giving up.
Let $\boldsymbol{\theta}_o$ be the initial parameters.
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_o$
$i \leftarrow 0$
$j \leftarrow 0$
$v \leftarrow \infty$
$\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$
$i^* \leftarrow i$
while $j < p$ **do**
  Update $\boldsymbol{\theta}$ by running the training algorithm for n steps.
  $i \leftarrow i + n$
  $v' \leftarrow \text{ValidationSetError}(\boldsymbol{\theta})$
  if $v' < v$ **then**
    $j \leftarrow 0$
    $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$
    $i^* \leftarrow i$
    $v \leftarrow v'$
  else
    $j \leftarrow j + 1$
  end if
end while
Best parameters are $\boldsymbol{\theta}^*$, best number of training steps is i^*.

[Goodfellow et al 2016, Algorithm 7.1]

Data Augmentation

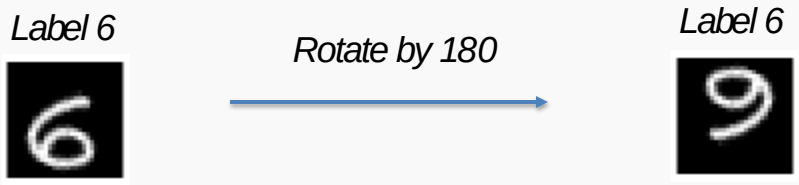
Data Augmentation is adding fake data to your training data set.



[CS M148 Winter 2024]

Data Augmentation: dos and don'ts

Carefully choose your transformations. Not all transformations are valid.



Data Augmentation does not work for tabular data and not as nicely for time series.

Works will for object recognition and speech recognition

[CS M148 Winter 2024]

Data Augmentation

What are some other means for data augmentation?

1. *Adding noise to data*
2. *Bootstrap*
3. *Gaussian Mixture Models (Here use as a generative model)*

Today's Learning Objectives

Students will be able to:

- ✓ Review Exercise: NN Training and Learning Rate
- ✓ Review: Regularization for NN
  - Ensembles and Dropout
  - Interpretability for NN

Random Forests

Random Forest (RF) algorithm is an **ensemble** algorithm that combines many decision trees:

- Training each tree using a different random bootstrap sample of the training data.

- Considering a different random subset of the predictor variables for each node split

[Yu, Barter 2024]

Random Forests Predictions

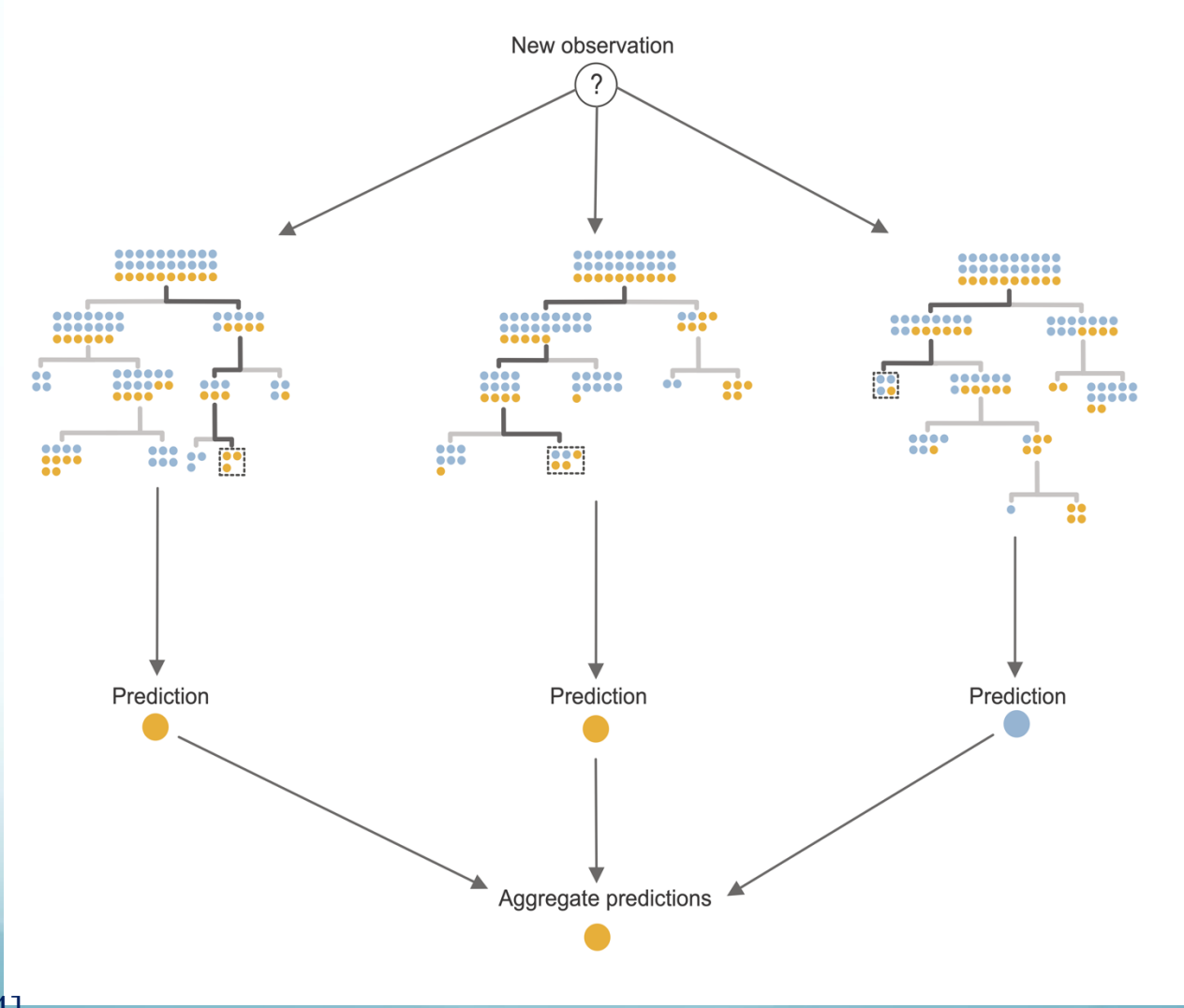
For binary response problems,
a **class probability prediction** can be computed based on the average probability prediction from the individual trees.

Binary class response predictions are computed either using a majority vote of the individual tree binary predictions or by applying a threshold to the class probability predictions.

A **continuous response prediction** can be computed by averaging the predicted responses across the individual trees.

[Yu, Barter 2024]

Random Forests Example

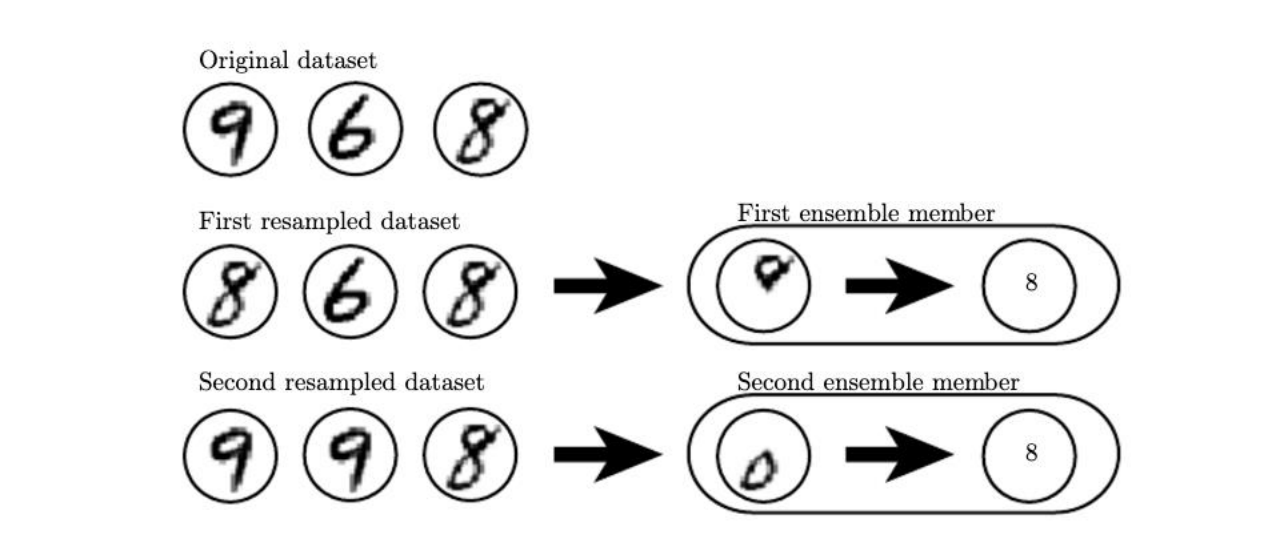


[Yu, Barter 2024]

Bootstrap Aggregating

Bootstrap aggregating or **bagging**: Train several different models separately on different bootstrapped samples and then have average the results

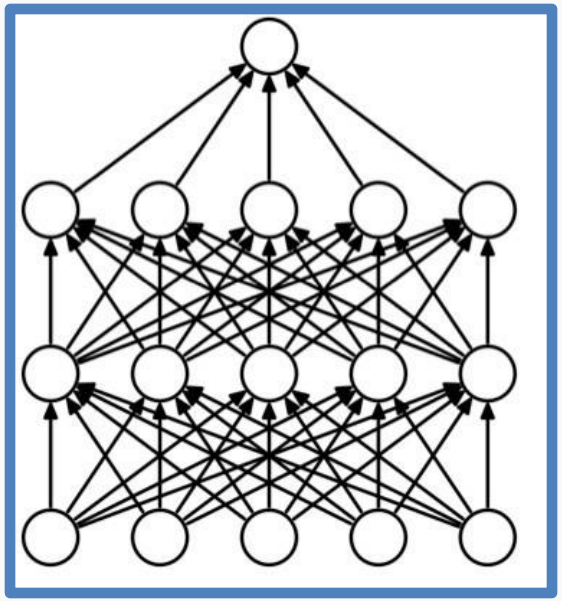
This is called **model averaging.**



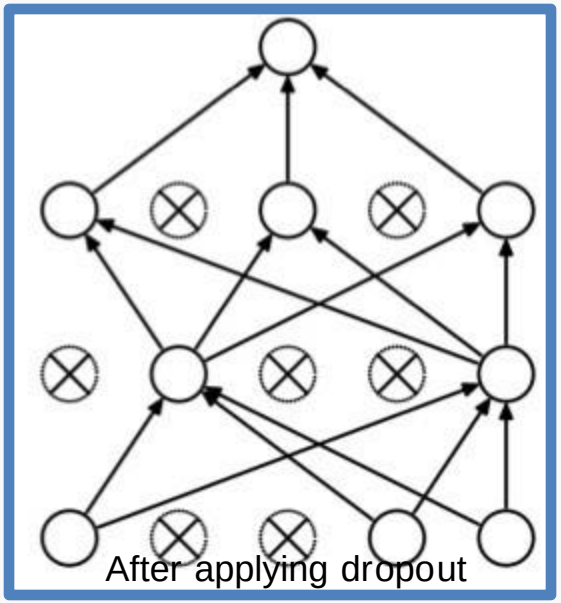
[Goodfellow et al 2016, Figure 7.5]

Dropout

- Proposed by Hinton (2012) and Srivastava et al (2014)
- Randomly set some neurons and their connections to zero (i.e. “dropped”)
- Prevent overfitting by reducing co-adaptation of neurons
- Like training many random sub-networks



Standard Neural Network



After applying dropout

https://arxiv.org/pdf/1207.0580.pdf

Dropout: Training

For each new example in a mini-batch (could be for one mini-batch depending on the implementation):

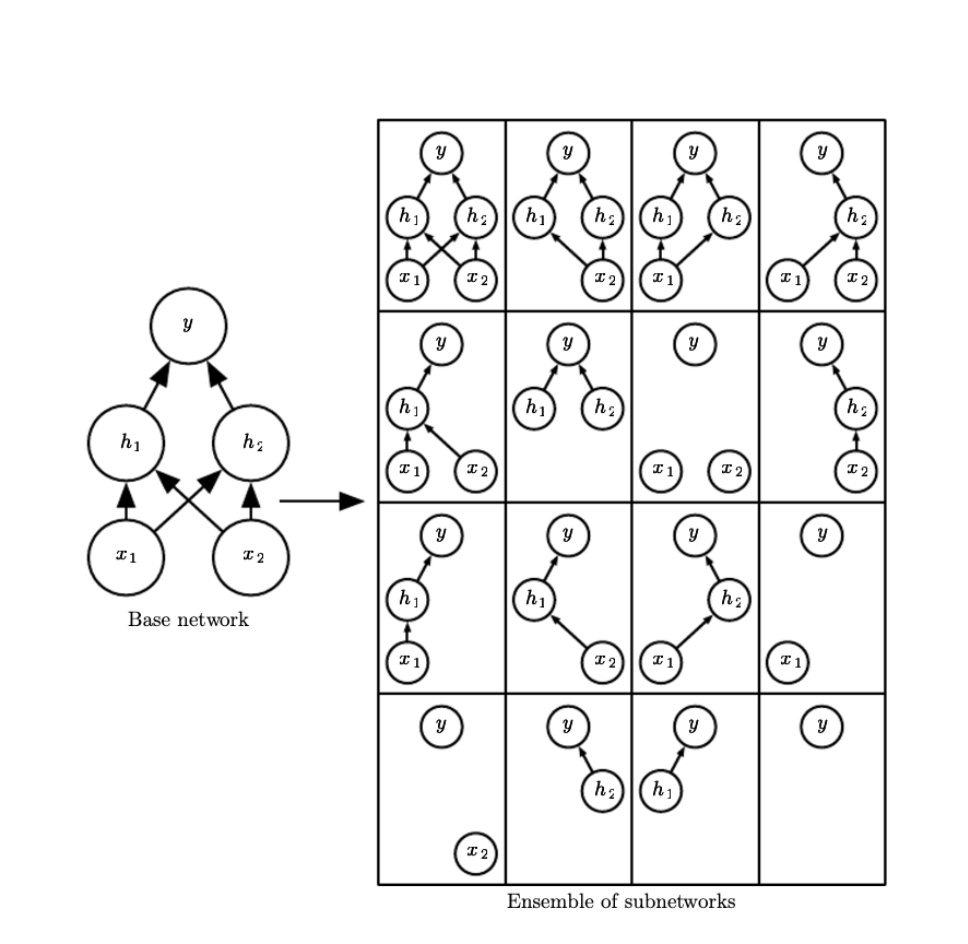
- Randomly sample a binary mask μ independently, where μ_i indicates if input/hidden node i is included
- Multiply output of node i with μ_i, and perform gradient update

Typically:

- Input nodes are included with prob=0.8 (as per original paper, but rarely used)
- Hidden nodes are included with prob=0.5

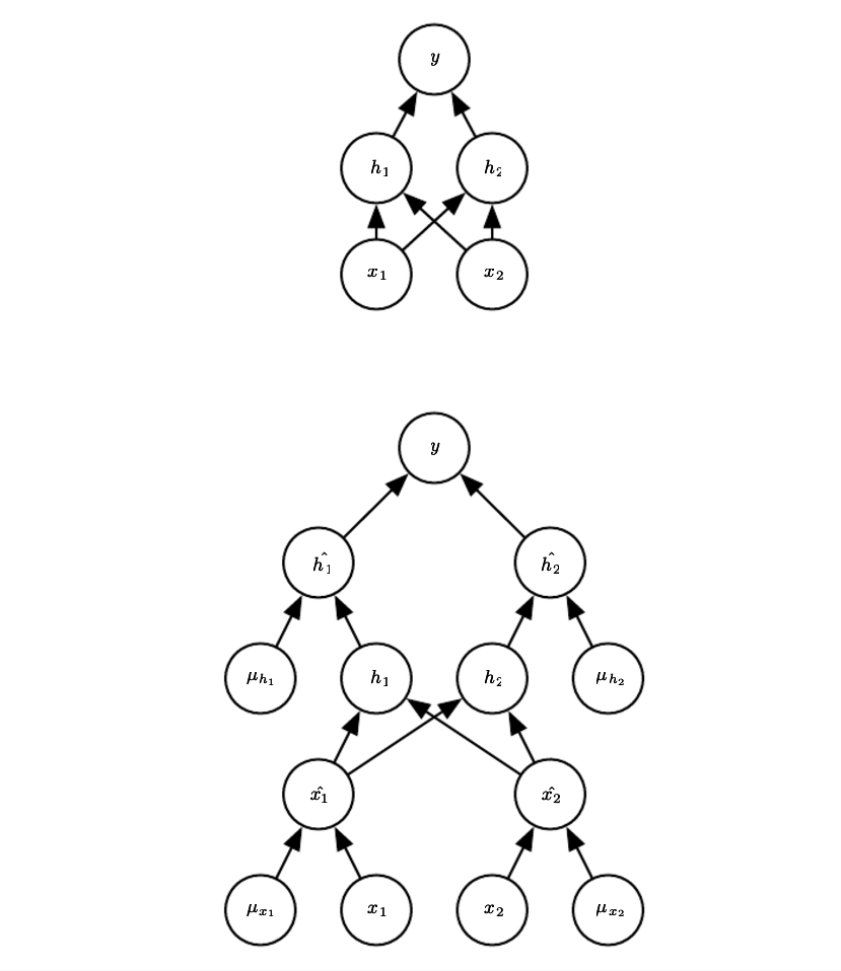
[CS M148 Winter 2024]

Dropout Trains from Exponential Set of models



[Goodfellow et al 2016, Figure 7.6]

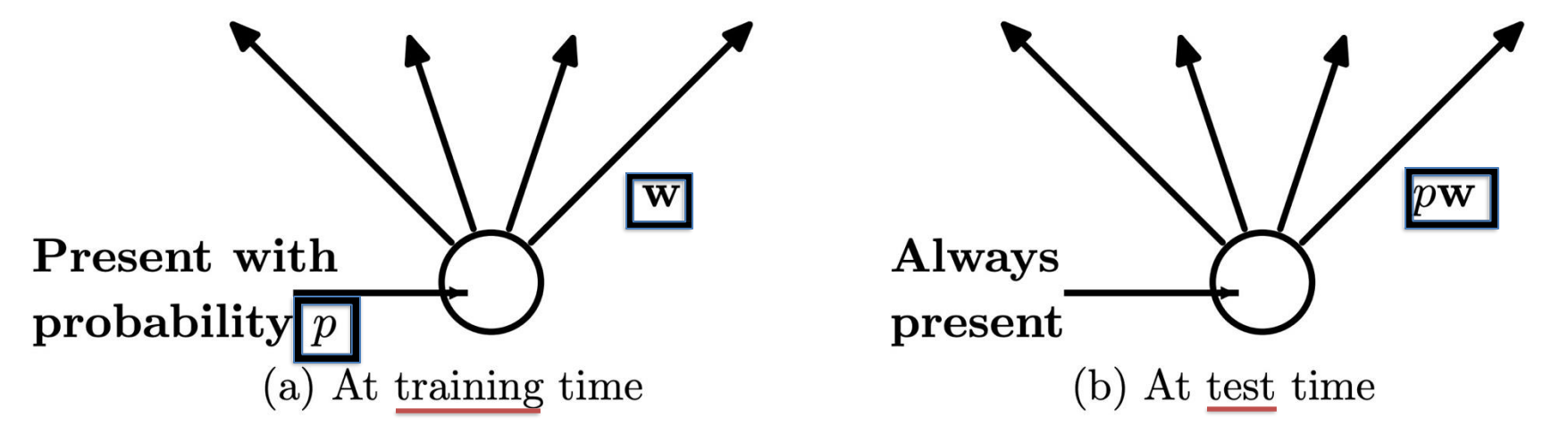
Dropout Forward Algorithm



[Goodfellow et al 2016, Figure 7.7]

Dropout: Prediction

- We can think of dropout as training many of sub-networks
- At **test time**, we can "aggregate" over these sub-networks by **reducing connection weights in proportion to dropout probability, *p***



NOTE: Dropouts can be used for **neural network inference** by dropping during predictions and predicting multiple times to get a distribution

Dropout vs Bagging

- Both are training ensembles of models
- In bagging each model is independent
- In bagging each model is trained on entire training set
- In dropout, the models share parameters!!
- In dropout, some set of models are sampled and trained on a single sample.
- In dropout, the parameter sharing across sampled models helps set parameters and make problem tractable.

- Otherwise both use bootstrapping and averaging of models.

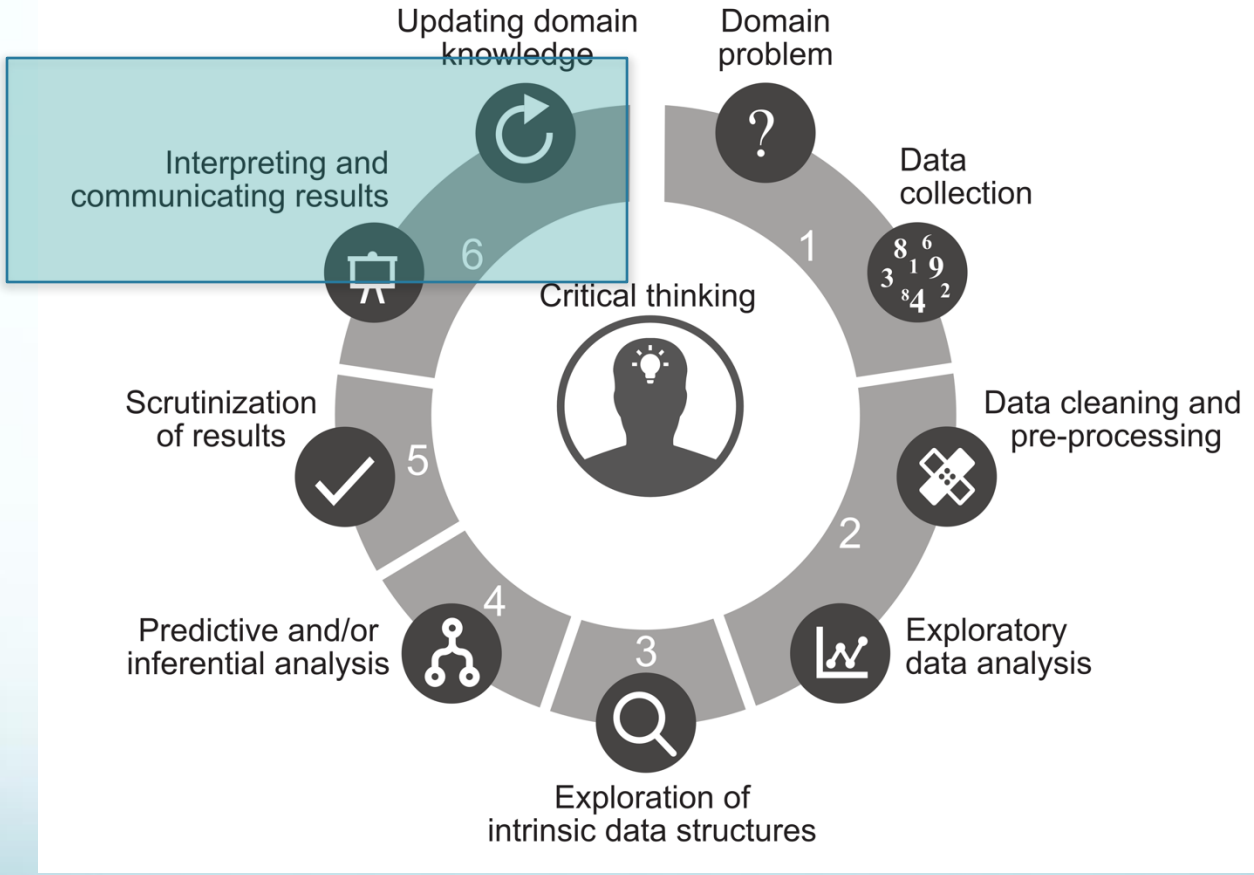
[Goodfellow et al 2016]

Today's Learning Objectives

Students will be able to:

- ✓ Review Exercise: NN Training and Learning Rate
- ✓ Review: Regularization for NN
- ✓ Ensembles and Dropout
- Interpretability for NN

Data Science Life Cycle (DSLC)



[Yu, Barter 2024]

Can we Trust Our Models?

When we train a model, we implicitly trust that the model make sensible predictions. Such assessment is usually done by looking at held-out accuracy or some other aggregate measure.

However, such metrics can be very misleading.

- *There can be data leakage*
- *The results may not generalize*
- *The results may not tell us what features are important to consider for further study*

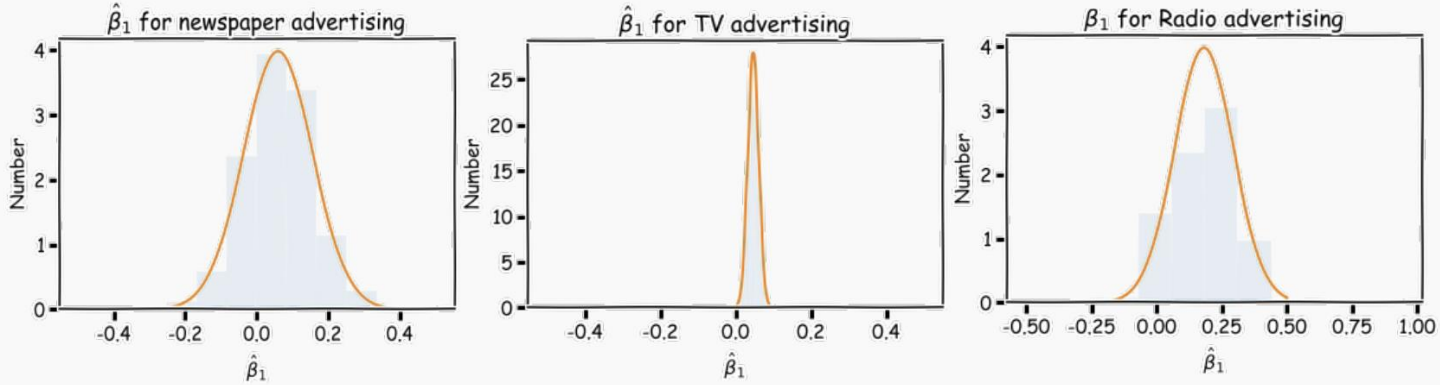
Understanding the model's predictions can be an additional useful tool when deciding if a model is trustworthy or not.

[CS M148 Winter 2024]

Feature importance

Now we know how to generate these distributions we are ready to answer ***two important questions:***

A. Which predictors are most important?
B. And which of them really affect the outcome?



$\mu_{\beta_1} = 0.03$
σ_{β_1}=0.13

$\mu_{\beta_1} = 0.033$
σ_{β_1}=0.01

$\mu_{\beta_1} = 0.23$
σ_{β_1}=0.25

0

Comparing coefficients

$$\text{predicted price} = b_0 + b_1\text{area} + b_2\text{quality} + b_3\text{year} + b_4\text{bedrooms}.$$

How do we compare the coefficients? Can we simply use how large some coefficients are compared to others?

[Yu, Barter 2024]

Comparing coefficients

The coefficients of different predictive features (predictor variables) are not comparable unless

- They're on the same scale

- Coefficients have been standardized to create **t-values:**

$$t_j = \frac{b_j}{SD(b_j)}.$$

[Yu, Barter 2024]

Creating t-values

1. Create N (e.g., $N = 100$ or $N = 1,000$) bootstrapped versions of the original dataset so that each of the N bootstrapped datasets has the same number of observations as the original data.

2. For each of the N bootstrapped datasets, compute an LS fit and extract the relevant coefficient value (so that you have N versions of each coefficient value).

3. Compute the SD of the N bootstrapped coefficients.

$$t_j^{\text{boot}} = \frac{b_j}{SD^{\text{boot}}(b_j)}.$$

[Yu, Barter 2024]

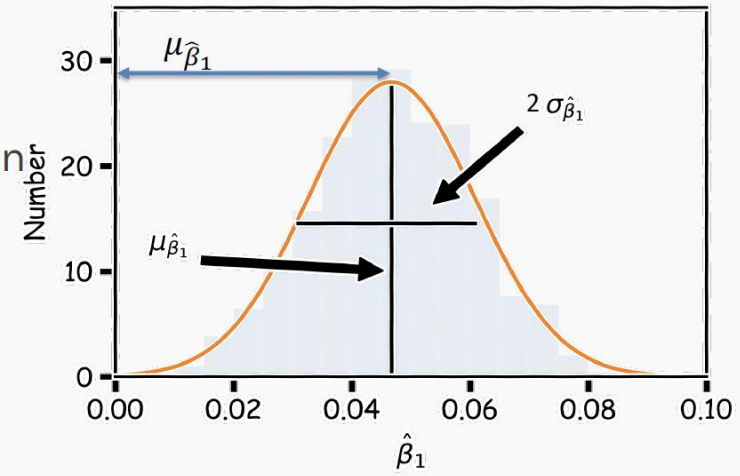
Feature Importance

To incorporate the coefficients' uncertainty, we need to determine whether the estimates of β's are sufficiently far from zero.

To do so, we define a new metric, which we call t-test statistic:

$$t = \frac{\mu_{\hat{\beta}_1}}{\sigma_{\hat{\beta}_1}}$$

which measures the distance from zero in units of standard deviation.



Feature Importance

How do we assess if there is a true relationship between outcome and predictors?

Compare its significance (t-test) to the equivalent measure from a dataset where we know that there is no relationship between predictors and outcome.

There will be no such relationship in data that are randomly generated. Therefore, we want to compare the t-test of the predictors from our model with t-test values calculated using random data. This is constructing a **null distribution.**

Hypothesis Testing

Hypothesis testing is a formal process through which we evaluate the validity of a statistical hypothesis by considering evidence **for** or against the hypothesis gathered by **random sampling** of the data.

1. State the hypotheses, typically a **null hypothesis** and an **alternative hypothesis** that is the negation of the former.
2. Choose a type of analysis, i.e. how to use sample data to evaluate the null hypothesis. Typically this involves choosing a single test statistic.
3. **Sample** data and compute the test statistic.
4. Use the value of the test statistic to either reject or not reject the null hypothesis.

Getting a null distribution: Permutation Testing

1 For *n* random datasets fit *n* models. One easy way to do this is to permute the dependent variable n times.

2. Fit the model using the permuted dependent variables and original. The permuted dependent variables are creating the null distribution.
3. Calculate the t-tests.

P-value

To compare the t-test values of the predictors from our model, $|t^*|$, with the t-tests, calculated using random data, $|t^R|$, we estimate the probability of observing $|t^R| \geq |t^*|$.

We call this probability the p-value.

$$p - value = P(|t^R| \geq |t^*|)$$

We call this probability the p-value.

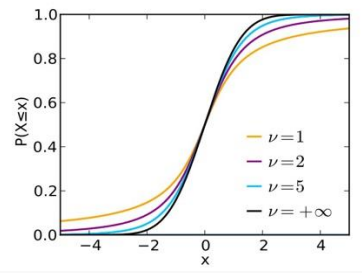
$$p - value = P(|t^{\cdot}| \geq |t^*|)$$

small p-value indicates that it is unlikely to observe such a substantial association between the predictor and the response due to chance.

It is common to use p-value<0.05 as the threshold for significance.

To calculate the p-value we use the cumulative distribution function (CDF) of the student-t.

`stats model` a python library has a build-in function `stats.t.cdf()` which can be used to calculate this.



Hypothesis testing

1. **State Hypothesis:**

Null hypothesis: There is no relation between *X* and *Y*

The alternative: There is some relation between *X* and *Y*

2. **Choose test statistics**

t-test

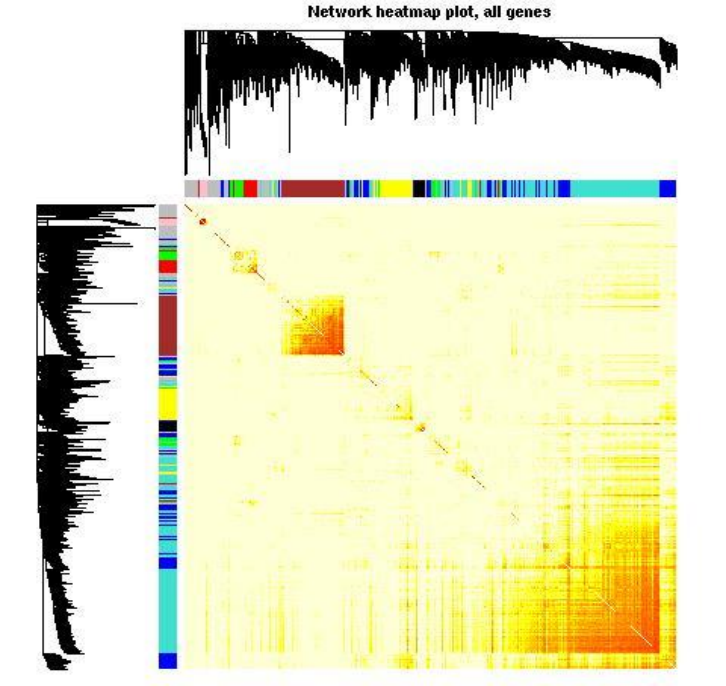
3. **Do permutation testing**

4. **Reject or not reject the hypothesis:**

We compute ***p-value ,*** the probability of observing any value equal to $|t|$ or larger, from random data.

p-value < p-value-threshold we reject the null.

Finding Associations of biomarkers for diseases



- In Genome Wide Associate Studies (GWAS) we may be given millions of markers known as single nucleotide polymorphisms or SNPs
- We want to know what markers may be associated with diseases such as diabetes or Alzheimer's. We may be given a patient's disease status or a quantitative measurement such as A1C. These we use as the phenotype for a disease.

Given a quantitative phenotype, p, and each SNP, g, GWAS use the following regression model:

$$p = \beta_0 + \beta_1 g$$

Once we fit the model, the coefficient for g is an estimate of its association with the phenotype.

Given a quantitative phenotype, p, and each SNP, g, GWAS use the following regression model:

$$p = \beta_0 + \beta_1 g$$

We use hypothesis testing: Our null hypothesis is that the coefficient for g is 0.

The probability of data given the null hypothesis is true is the **p-value.**

We want the p-value to be as small as possible to select g as a marker: *this means the data supports that the marker has a non-zero association with the phenotype. We want to limit false positives or Type I error.*

Can we Trust Neural Networks?

A doctor will certainly not operate on a patient simply because "the model said so."

Modern machine learning models are obscure!

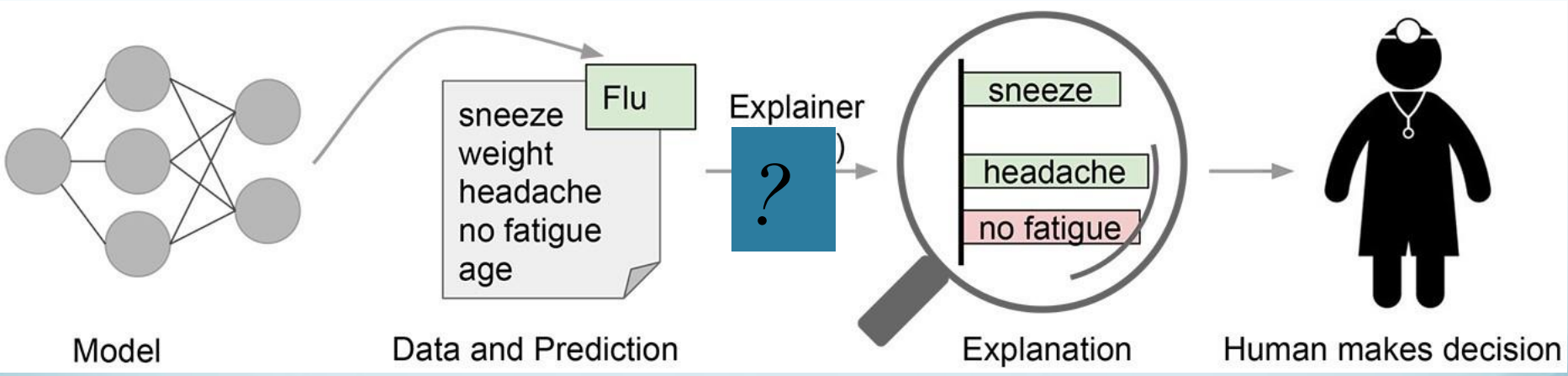
- *Understanding the rationale behind the model's predictions would help users decide when to trust or not to trust their predictions.*

How can we explain the prediction of models such as neural networks?

Can we Trust Neural Networks?

Explain the predictions postdoc:

- *Imagine prediction of flu is explained by an "explainer" that highlights the symptoms that are most important to the model.*
- *With this information about the rationale behind the model, the doctor is now empowered to trust the model—or not.*



[CS M148 Winter 2024]

Shapley values

Let's take as example a critical use case that directly involve us in our lives: ***Credit Risk Scoring****.*

Bob wants to buy a house and he wants to ask the bank for a loan.

The bank will use Bob's data in their internal model to see if they should grant the loan.

The bank's model will calculate the ***probability of default on loan.***

If the bank refuses Bob's request, Bob may ask for justification.

Shapley values

The Banks simplified model uses only 3 features: the income, the location and the age.

How can we understand how much each feature value contributed to the final prediction?

Shapley values *are from game theory:*
In a group of players with different skill sets that worked together to reach a payout, what is the fairest way to split the payout among them?

In machine learning: the game is the prediction task, the "payout" is model output and the "players" are the features.

Shapley values

Bob's data: 29 years old, his annual gross income is $25k and he lives in Milan.

Average default rate is 40%

Each feature contributes an additive 15% chance of default

Income is a particular driver and affects 80% probability of default when considered with at least one other feature (composing what is usually called a coalition).

If the model output by considering all 3 features is an 80% of probability of default, how much did each one contribute to this result?

Shapley values

Let's look at an example of coalition of features with and without age

- *We should compute each feature's **marginal contribution**, averaged over every possible sequence in which the features could have been added to the coalition. So in this case we have 4 possible coalitions:*
  1. *{}*
  2. *{income}*
  3. *{location}*
  4. *{income, location}*

- *For each coalition, we determine the marginal contribution by taking the difference between the model output with and without the feature for which we are computing the contribution*

- *Finally, we average all of these payoffs together, and we have the Shapley values for our feature age.*

Shapley values

Let's look at an example of coalition of features with and without age

	coalition_without_age	PoD_without_age	coalition_with_age	PoD_with_age	PoD_difference
2	{}	40%	{age}	55%	15%
3	{income}	55%	{income, age}	80%	25%
4	{location}	55%	{location, age}	70%	15%
5	{income, location}	80%	{income, location, age}	80%	0%

[CS M148 Winter 2024]

Shapley values

- Notice that the order of the features in the set does not matter!

- We can weight the marginal contributions by the number of permutations of the coalition

- We'll use the **multinomial coefficient:**

Numerator: The number of permutations of coalition, ***S***, multiplied by the number of permutations of features left to be added excluding current feature and coalition

Denominator: Total number of permutations of all features

Shapley values

In a formula, letting v being the payout or value function, the Shapley Value for the feature i can be computed as:

$$\phi_i(v) = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!}[v(S \cup i) - v(S)]$$

where v(S) is the prediction for feature values in set S that are marginalized over features that are not included in set S minus the average prediction:

[CS M148 Winter 2024]

Shapley values

Let's calculate multinomial coefficient:

What is the multinomial coefficient for {} and {income, location}?

What is the multinomial coefficient for for {income} and {location})?

Shapley values

What is the Shapley value for age?

	coalition_without_age	PoD_without_age	coalition_with_age	PoD_with_age	PoD_difference
2	{}	40%	{age}	55%	15%
3	{income}	55%	{income, age}	80%	25%
4	{location}	55%	{location, age}	70%	15%
5	{income, location}	80%	{income, location, age}	80%	0%

$$\phi_i(v) = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [v(S \cup i) - v(S)]$$

[CS M148 Winter 2024]

Shapley values

Evaluating all possible coalitions of features ***exponentially increases*** *with the number of features.*

A general simple approach for the ***approximate Shapley estimation for a single value*** *with a pseudo-algorithm:*

Algorithm 1 Approximate Shapley estimation for a single value

Require: Number of iterations M, instance of interest x, feature index j, training set X, ML model f

Ensure: Shapley value for the value of feature j

for $m = 1, ..., M$ **do**

Draw random instance z from the dataset X

Choose a random permutation o of the features values

Order instance x: $x_o = (x_{(1)}, ..., x_{(j)}, ...x_{(p)})$

Order instance z: $z_o = (z_{(1)}, ..., z_{(j)}, ...z_{(p)})$

Construct two new instances

With j: $x_{+j} = (x_{(1)}, ..., x_{(j-1)}, x_{(j)}, z_{(j+1)}, ...z_{(p)})$

Without j: $x_{-j} = (x_{(1)}, ..., x_{(j-1)}, z_{(j)}, z_{(j+1)}, ...z_{(p)})$

Compute marginal contribution: $\phi_j^m = f(x_{+j}) - f(x_{-j})$

end for

Compute Shapley Value as the average: $\phi_j(x) = \frac{1}{M} \sum_{m=1}^{M} \phi_j^m$

Shapley values

Note that for approximation purposes,

1. *Sampling random instance z from the data and a random order of features.*
2. *The features of z are used to "fill" the missing features that are not in the coalition (those ones after j) as well as the specific feature j in the second instance x_j.*
3. *Average all the differences, and this implicitly weights samples by the probability of the distribution X.*
4. *This procedure is repeated for each of the features to get all Shapley Values.*

Today's Learning Objectives

Students will be able to:

- ✓ Review Exercise: NN Training and Learning Rate
- ✓ Review: Regularization for NN
- ✓ Ensembles and Dropout
- ✓ Interpretability for NN

Citations:.

Goodfellow, I., Bengio, Y.,, Courville, A. (2016). *Deep Learning*. MIT Press.

Sebastian **Raschka**, Yuxi (Hayden) **Liu**, and Vahid Mirjalili. **Machine Learning** with PyTorch and Scikit-Learn. Packt Publishing, **2022**.

Thank You