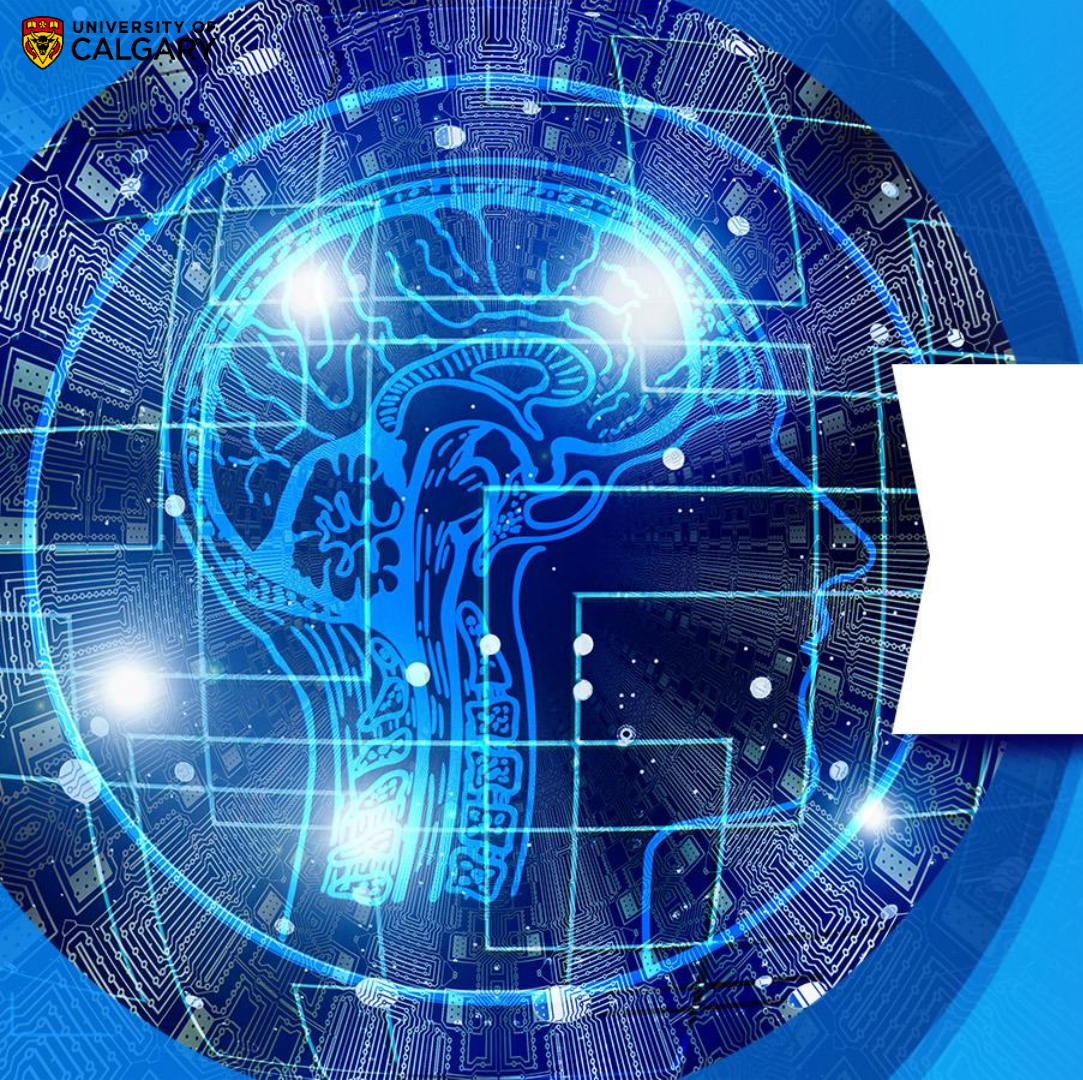


Transfer Learning

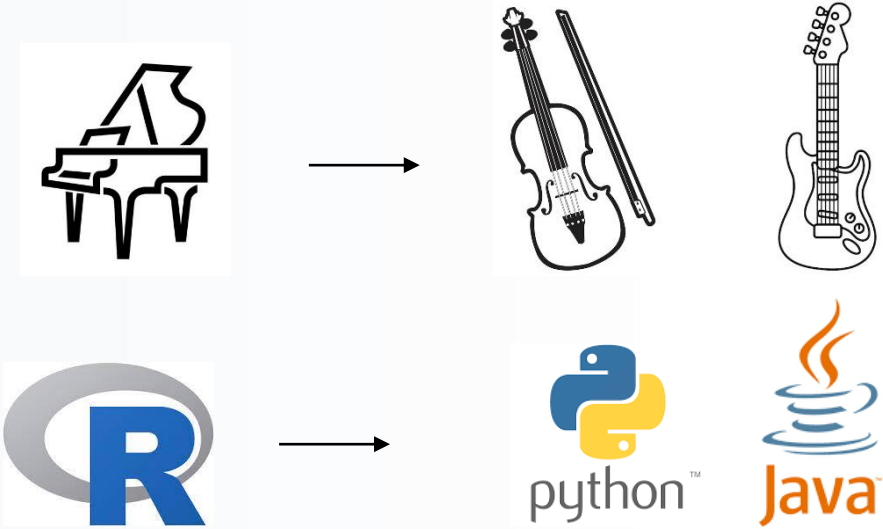
Qing(Leah) Li
Supervised by Dr. Quan Long
Cumming School of Medicine
University of Calgary

Outlines

- Part I:
  - 1. What is transfer learning?
  - 2. Rationality for transfer learning
  - 3. Transfer learning examples
    - 3.1 Image classification
    - 3.2 DTL: Disease-specific variants detections (Leah's research project)
- Part II:
  - 4. Deep learning basic glossary
  - 5. Hands-on: Image classification
    - Google colab

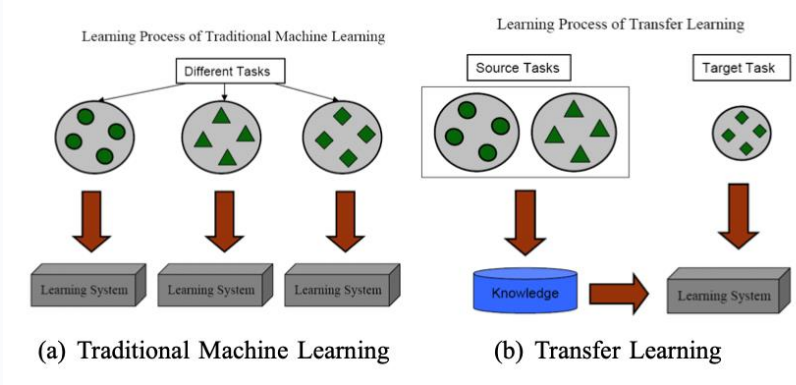
1. What is transfer learning?

- Everyone does transfer learning since very little



Transfer learning focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [wiki]

2. Rationality for transfer learning



- Advantages:
  - Some pre-trained models (giants)
    1. Are proven to be successful and valuable
    2. Large-scale architectures
    3. Trained on large datasets (like ImageNet)
    4. Are refined many times
    5. Are trained with many GPU resources
  - Transfer learning (standing on the shoulders of giants)
    1. Transfer the success (or knowledge) from pre-trained models
    2. Less errors adjusting architecture
    3. Training is easier and faster

Image source: Pan & Yang, IEEE Trans Knowledge & Data Engineering 2009

UNIVERSITY OF CALGARY

3. Transfer learning examples

- 3.1 Image classification
- 3.2 DTL: Disease-specific variants detections (Leah's research project)

3.1 Image classification

- A very conventional task
- Traditional machine learning:
  - Train from scratch

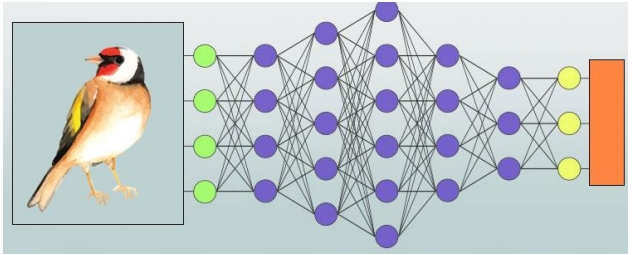


Image sources: https://www.educba.com/tensorflow-image-classification/

3.1 Image classification

- Modern transfer learning:
  - Convolutional neural network,
  - Using a pre-trained model to solve a new similar problem

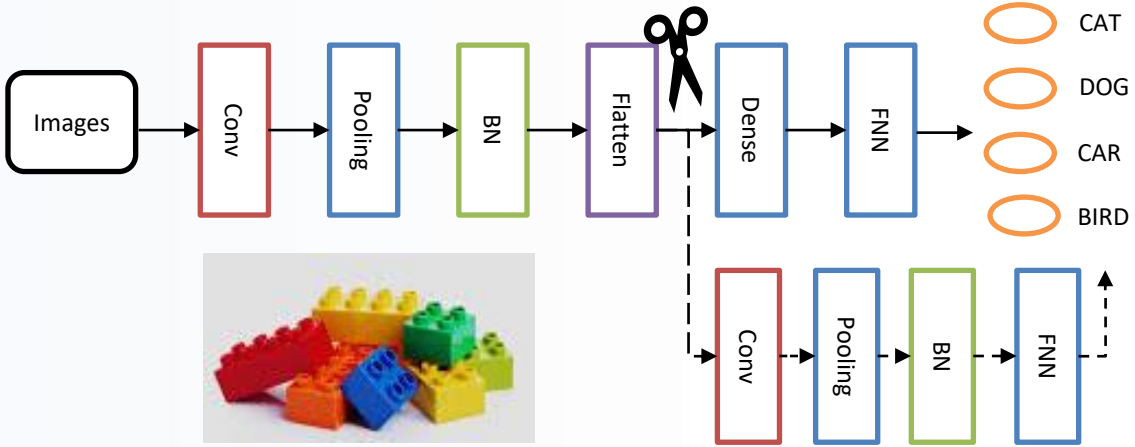
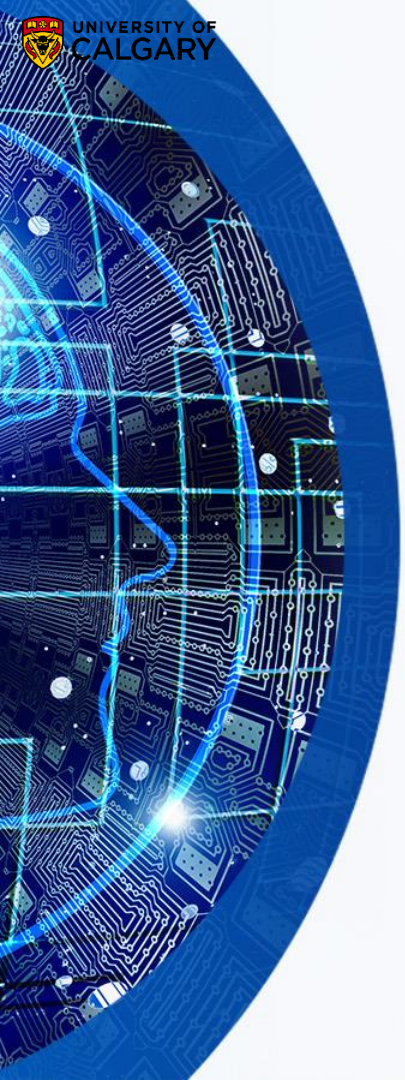


Image sources: https://www.educba.com/tensorflow-image-classification/



3.1 Image classification

- Tips for a successful transfer learning

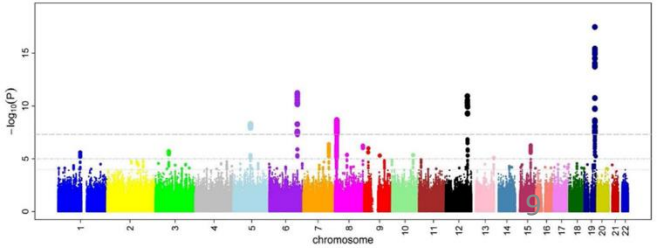
1. Choose an appropriate pre-trained model
   - Type of the application (image-based, text-based, …)
   - Scale of the pre-trained model
2. Compatibility at both the input and output ends of the base model
   - Shape
   - Data type

- Where to get pre-trained models?

1. Pre-trained models from other's (https://github.com/)
2. Tensorflow Hub (previously Model Zoo) pre-trained models, (https://tfhub.dev/)

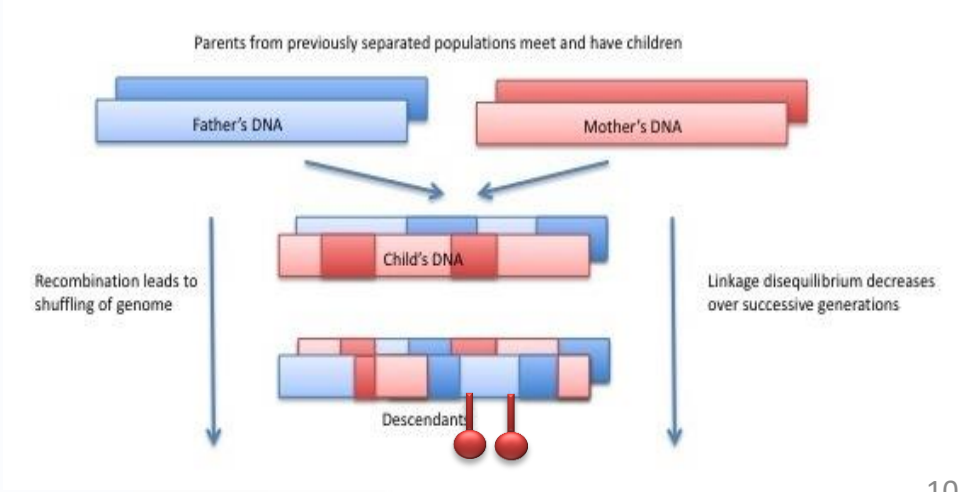
3.2 DTL: Disease-specific variants detections

- Genome-wide association studies (GWAS) aim to identify genetic basis that affects changes in phenotypes/traits.
- The simplest way is to conduct a linear regression : $y \sim x$
- **Association does not mean causation. How to identify real causal SNPs?**
- **Fine-mapping** is the process by which a trait-associated region from a genome-wide association study (GWAS) is analyzed to identify the particular genetic variants that are likely to causally influence the examined trait.



3.2 DTL: Disease-specific variants detections

- Why is it hard to determine which one is causal?
  - **linkage disequilibrium** (**LD**) is the non-random association of alleles at different loci in a given population.

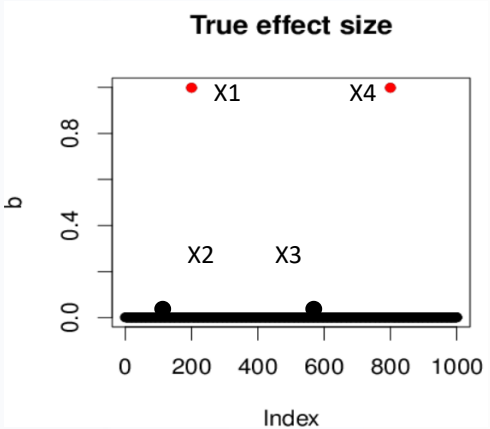


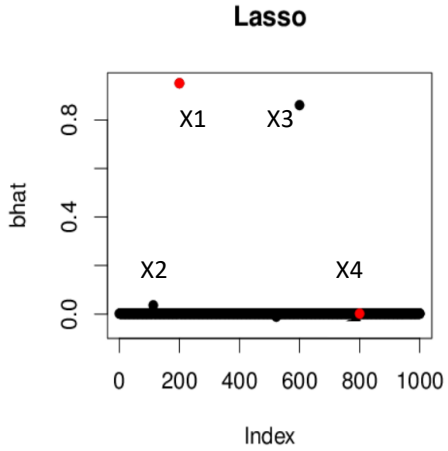
3.2 DTL: Disease-specific variants detections

The type of inference is motivated by the so-called "genetic fine-mapping" study. Consider fitting the regression model

$$y = \sum_{j=1}^{p} x_j \beta_j + \epsilon \quad \epsilon \sim N(0, \sigma^2 I_n)$$

where $x_1 = x_2, x_3 = x_4$ and $\beta_1 \neq 0, \beta_4 \neq 0, \beta_{j \notin \{1,4\}} = 0$. Our goal is to make a statement that

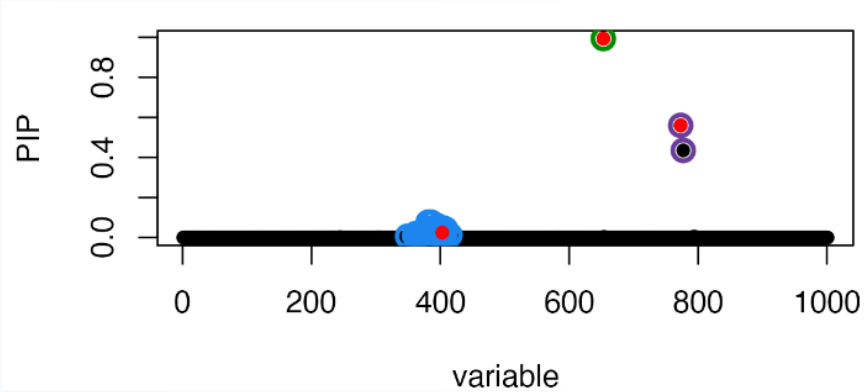




How about a group containing a causal SNP? {1,2}, {3,4}

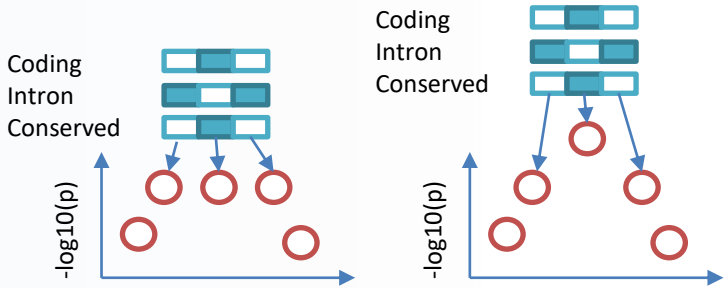
3.2 DTL: Disease-specific variants detections

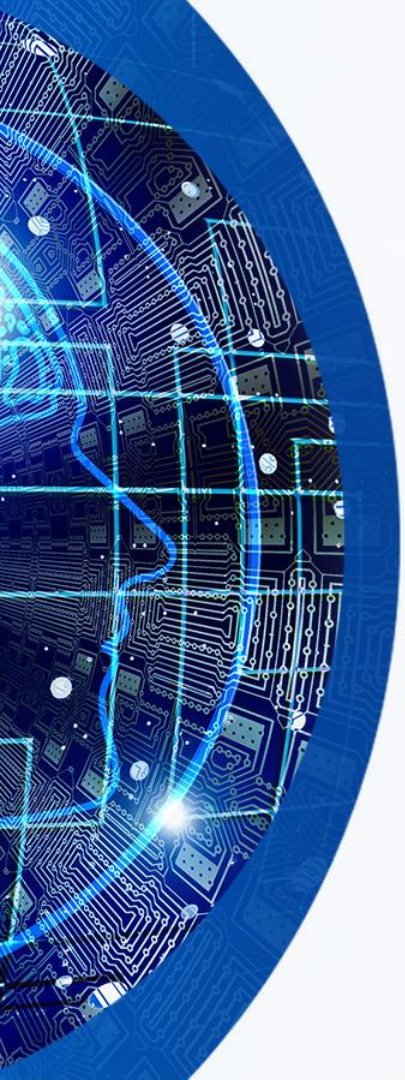
- **Smart idea: Credible Sets**
  - Definition: A 95% Credible Sets (CS) of variables is a set that has > 95% probability of containing at least one non-zero effect size
  - Hope to report as many CSs as possible, each as small as possible.
- **SuSiE**: Sum of Single Effects.
  - Input: individual genotype or GWAS summary statistics
  - Output: a number of “Credible Sets” (CSs), which highly likely contain a variable with non-zero effects, meanwhile being as small as possible. PIP: posterior inclusion probability



3.2 DTL: Disease-specific variants detections

- Limitation of SuSiE: it gives the same prior probability to all SNPs
  - Nonsynonymous mutations >> synonymous
  - Coding region mutations > noncoding regions mutations
- **Polyfun**: weight SNPs by their functional biological annotations
  - Input: hundreds/thousands of functional biological annotations
  - Output: per-SNP heritability, which can be regarded as the prior causal probability

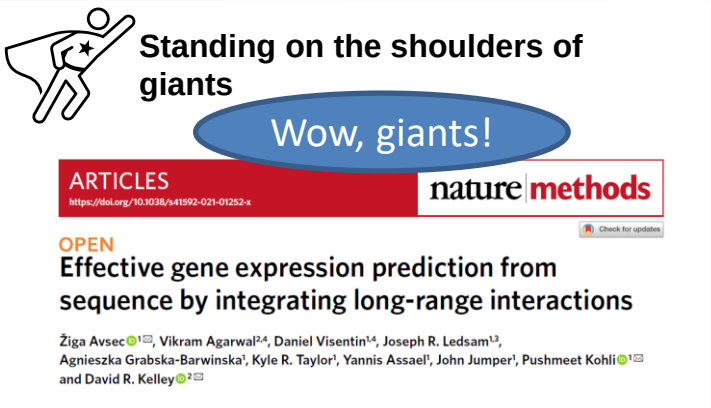




3.2 DTL: Disease-specific variants detections

- Goal: weight SNPs by their disease-specific functional biological annotations to identify more causal genetic variants affecting risk of human cancers.
- Approach: develop novel models through Deep Transfer Learning (DTL)

3.2 DTL: Disease-specific variants detections

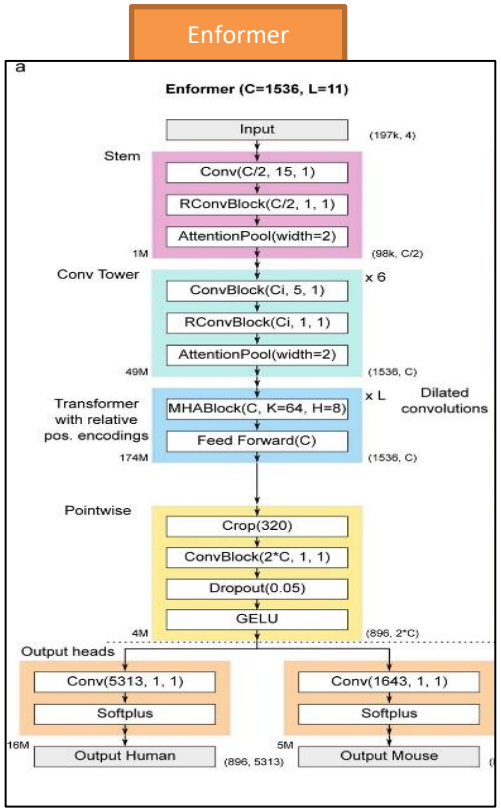


Enformer outputs tracks :

- **5,313** human genomic tracks (638 CAGE, 684 DNase/ATAC, and 3991 ChIP datasets)
- **1,643** mouse genomic tracks (357 CAGE, 228 DNase/ATAC, and 1058 ChIP datasets)

Enformer inputs DNA sequences:

- Human genomes: **38,171** (34,021 training, 2,213 validation, and 1,937 test) sequences
- Mouse genomes: **33,521** (29,295 training, 2,209 validation, and 2,017 test) sequences



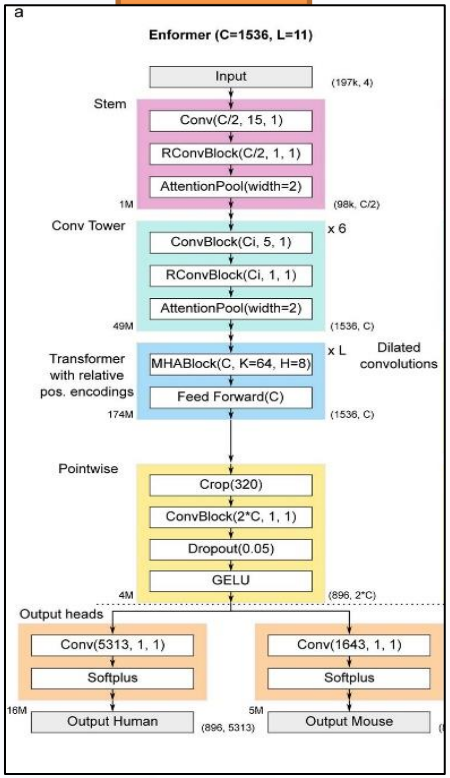

Qing (Leah) Li

Quan Long

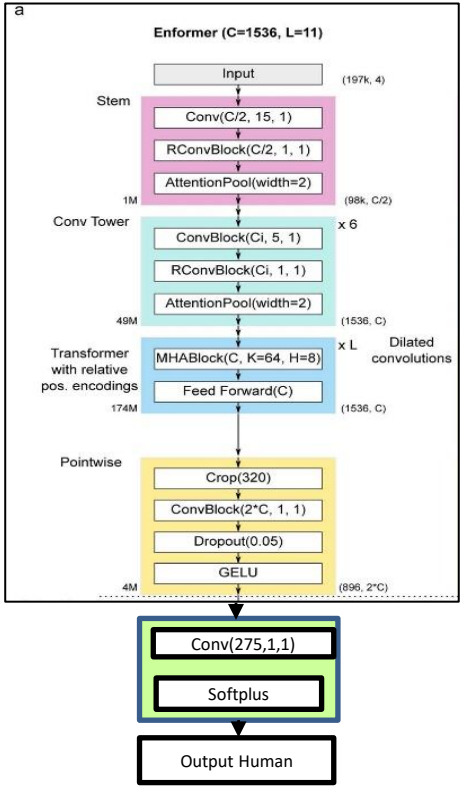
Xingyi Guo

3.2 DTL: Disease-specific variants detections

Enformer

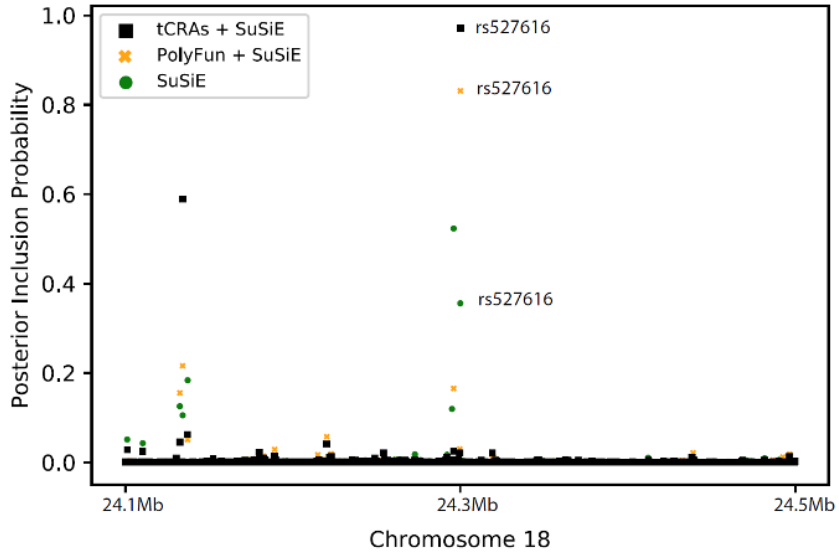


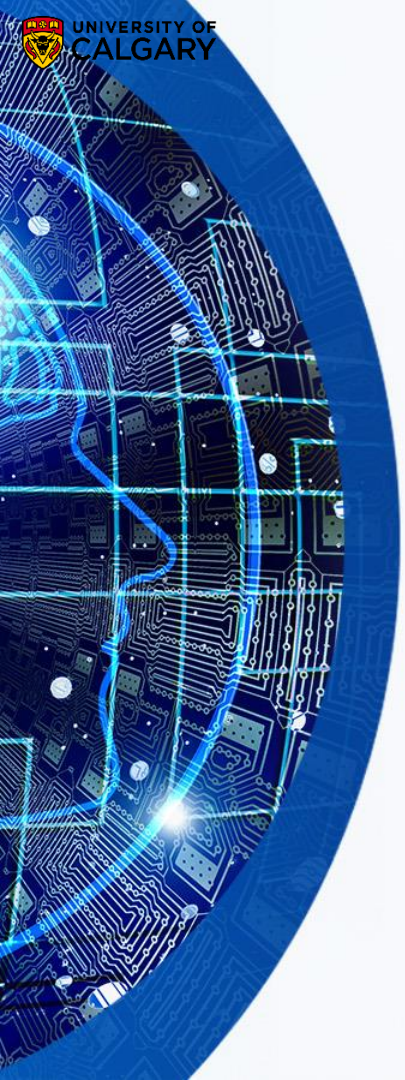
TFEnformer



3.2 DTL: Disease-specific variants detections

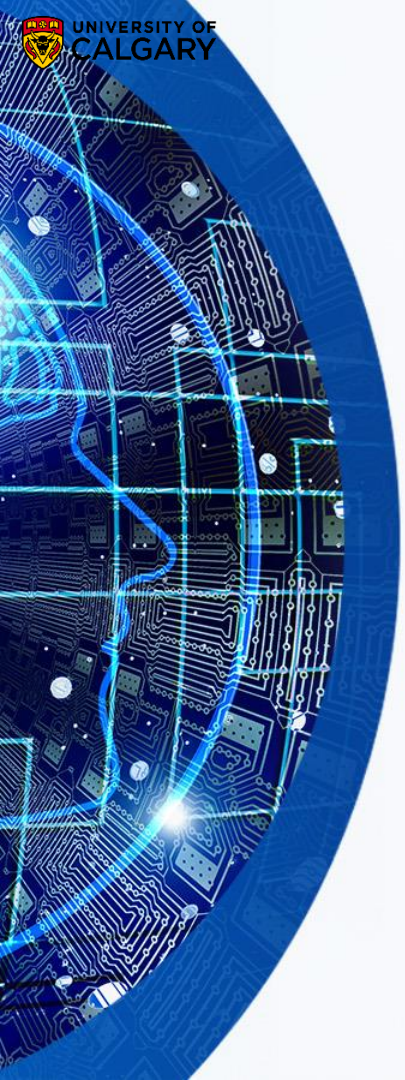
Fig. C1.3 Comparison of PIP for a putative casual variant, rs527616, identified by our fine-mapping approach, tCRAs+SuSiE, with other two approaches (PolyFun+SuSiE and SuSiE).





3.2 DTL: Disease-specific variants detections

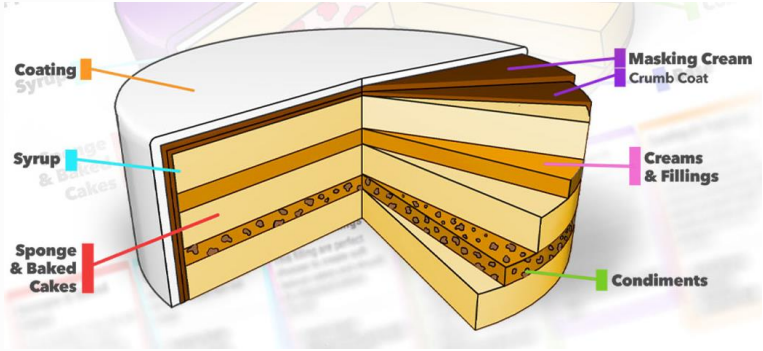
- Take away messages:
  - Functional biological annotations are important in discovering causal disease variants
  - However, disease-specific functional annotations were not integrated in previous models
  - Using transfer learning can fill up these gaps, thus promoting discoveries of disease-specific causal variants.



4. Deep learning basic glossary

1. Models' architectures
2. Dataset
3. Model training
   - Hyperparameters
   - Parameters
   - Epoch
   - Batch sizes
   - Iterations
   - Loss function
   - Feed forward and backpropagation
   - Gradient descent (Adam, SGD)
   - Learning rate
4. Model evaluation

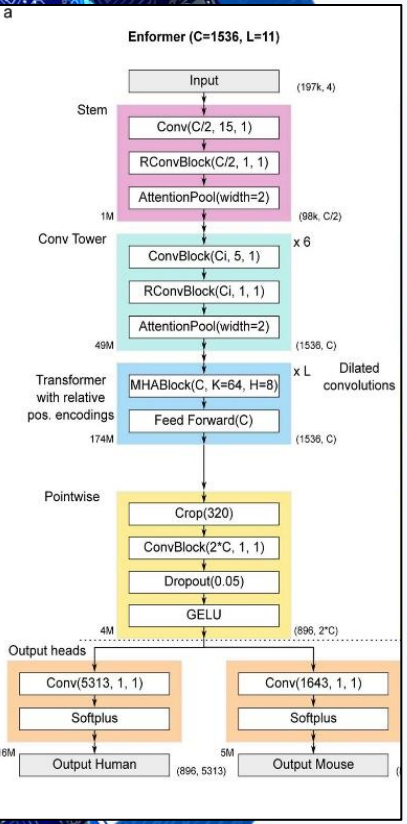
4.1 Models' architectures



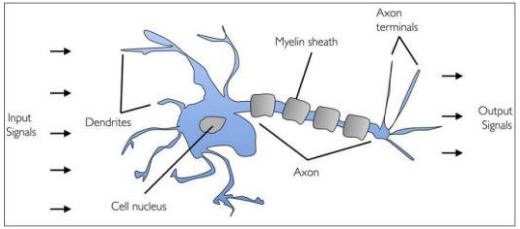
Sponges & Baked Cakes	Syrups	Creams & Fillings	Condiments	Masking Creams	Coating
The following choices of sponges and baked cakes can be used specially for the needs of wedding and novelty cakes, simply because they are more dense and can be kept in room temperature for a day. Even if these cakes get a bit softer in room temperature, they are still able to be portioned without destroying the layers and textures. Multiple choices can be used in one cake.	Brushing light syrups over the sponges can have a boosting effect in taste and also help to keep the cakes moist for a longer time. The ratio is 1:1 (stock syrup:water). Keep in mind that some cakes may not need extra moisture and using too much syrup can couse seeping/leaking and/or collapse of cake.	All the creams in this list can be reconditioned by slight heating in a microwave to a comfortable creamy consistency in order to sandwich/layer and/or mask cakes but most importantly to get a clean sliceable texture after the cake is kept in the fridge for a while.	The items in this list are some of the easy produced small crunchy bites or surprising flavours which can be easily added to the surface of creams or fillings. These items should be produced and stored in containers before hand. Condiments can be a great way to add value to the quality of a cake.	For some cases you can choose different components for fillings, crumb coating and masking purposes. Smoothness and colour is more important on the surface (masking) than it is for internal fillings. What masking cream is used may depend on what the final coating will be, for example, ganache will work well under fondant.	This is about the final surface of the cake. These items will effect the final aesthetics of the cake. Rolling, masking, rough paletting, pouring and piping are some common techniques which will be dependant on what component is chosen.
• Butter Cake	• Vanilla Syrup • Lemon Syrup • Mocha Syrup • Coconut Syrup • Whisky	• Butter Cream • Dark Chocolate Ganache • White Chocolate Ganache	• Caramelized Almond Crackling • Caramelized Walnut Crackling • Caramelized	• Butter Cream • Dark Chocolate Ganache	• Butter Cream • Dark Chocolate Ganache • White Chocolate Ganache • Caramel White Chocolate Ganache

https://www.yenersway.com/resources/combination-guide-for-layered-cakes/

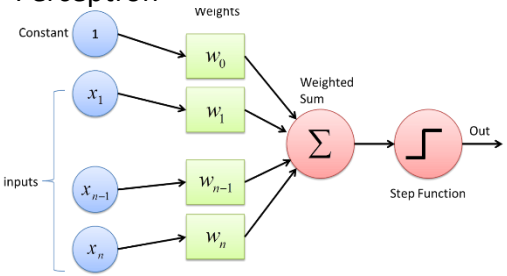
4.1 Models' architectures

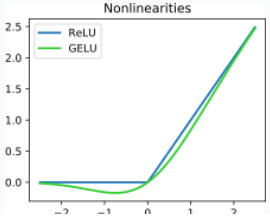


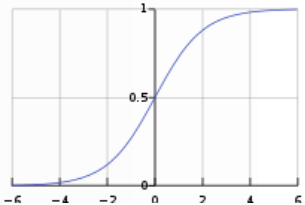
- Model
  - Functional blocks
    - Layers (neural network)
      - Fully connected
      - Sparsely connected (dropout)
      - Convolutional layers
      - Activation function (add nonlinearity to NN)



Perceptron







- ReLU activation: $\phi(\mathbf{v}) = \max(0, a + \mathbf{v}'\mathbf{b})$,
- Logistic activation: $\phi(\mathbf{v}) = (1 + \exp(-a - \mathbf{v}'\mathbf{b}))^{-1}$.

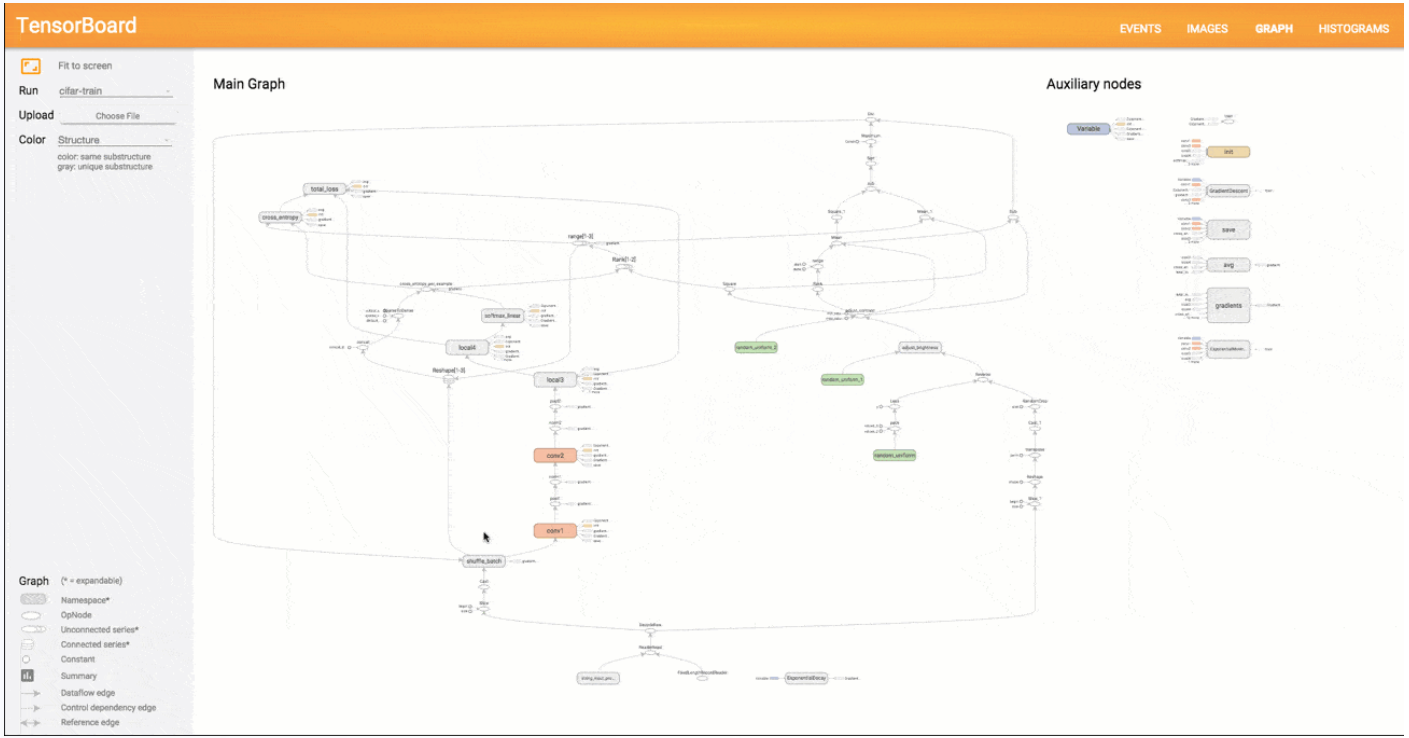
- Input and output shape

https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53; https://arxiv.org/pdf/1809.05165.pdf; https://www.simplilearn.com/tutorials/deep-learning-tutorial/perceptron; https://en.wikipedia.org/wiki/Activation_function

UNIVERSITY OF CALGARY

4.1 Models' architectures

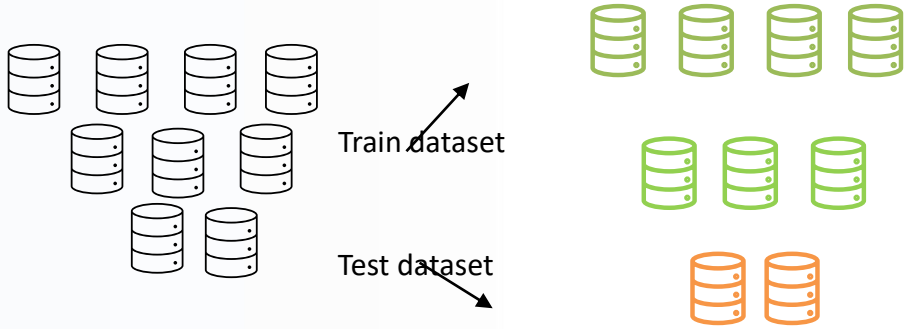
Visualization: TensorBoard, WANDB, etc.

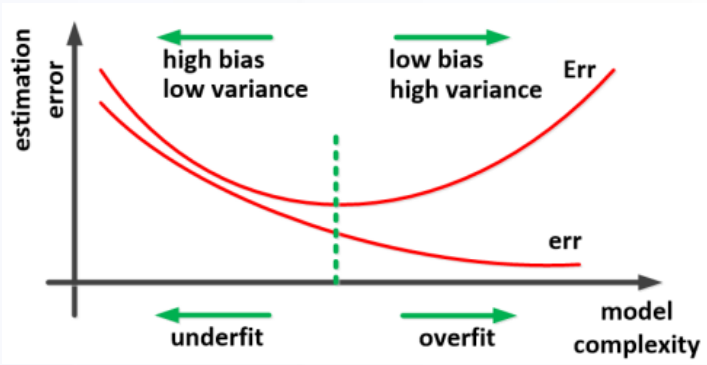


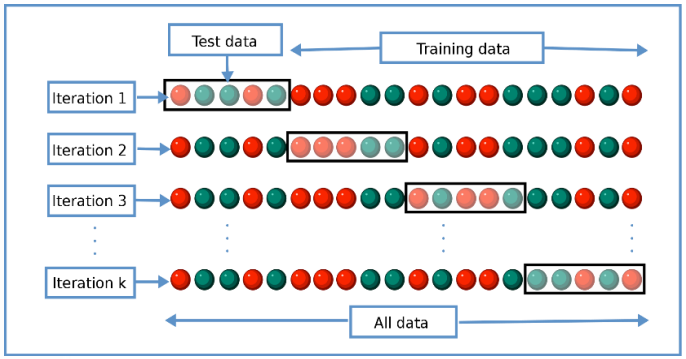
https://towardsdatascience.com/tensorboard-visualizing-learning-ad1b6667585

UNIVERSITY OF CALGARY

4.2 Datasets



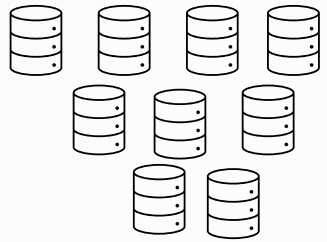


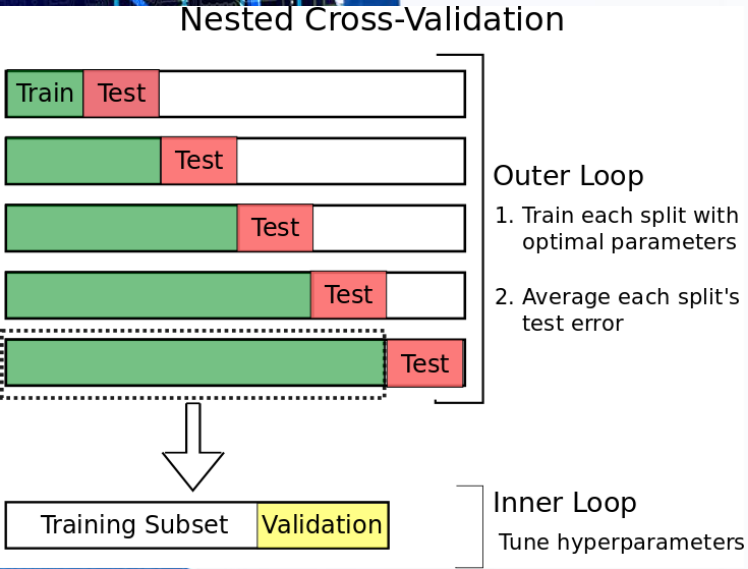


Train models' parameters on train datasets and measure models' performance on the test datasets

Ghojogh et al, arXiv 2019, https://en.wikipedia.org/wiki/Cross-validation_(statistics)

4.2 Datasets

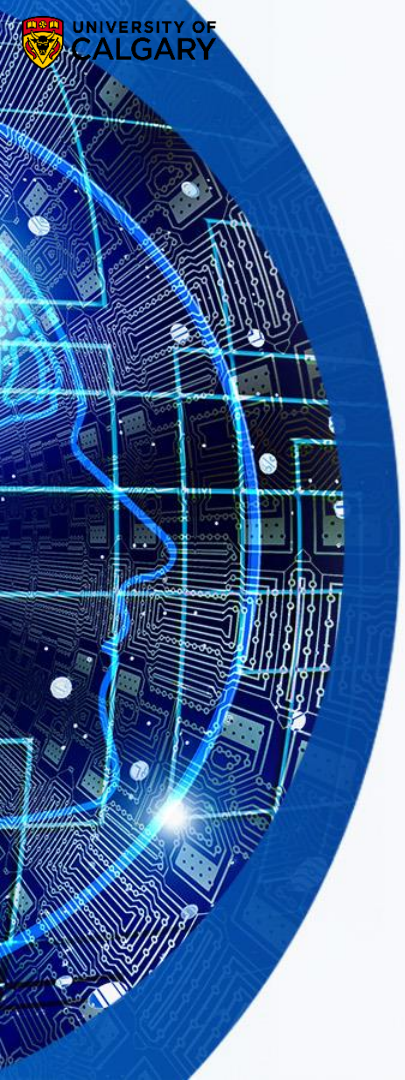




Nested cross-validation (double cross-validation) is an approach to model hyperparameter optimization and model selection that attempts to overcome the problem of overfitting the training dataset.

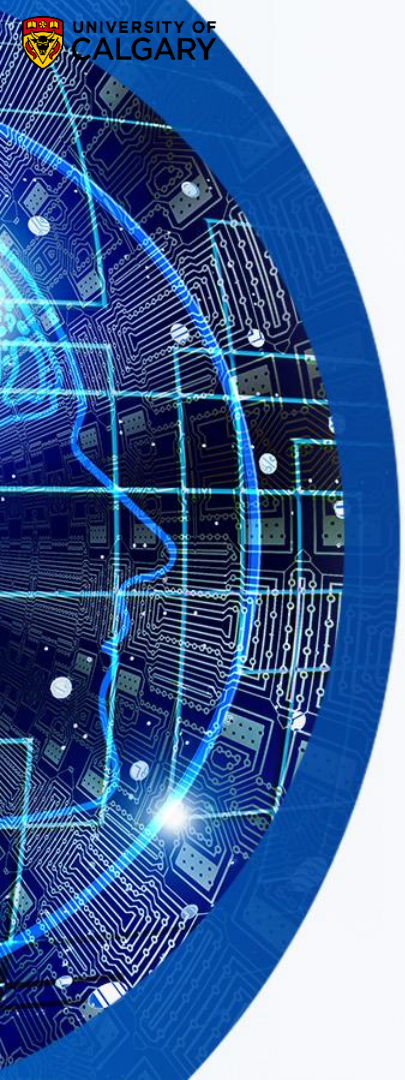
Cross-validation in Deep learning?

https://towardsdatascience.com/time-series-nested-cross-validation-76adba623eb9



4.3 Model training

- Hyperparameters
  - set manually and are used in processes to help estimate model parameters.
- Parameters
  - estimated from data automatically
- An example of hyperparameters from any of the lectures you learned so far?



Train dataset

4.3 Model training

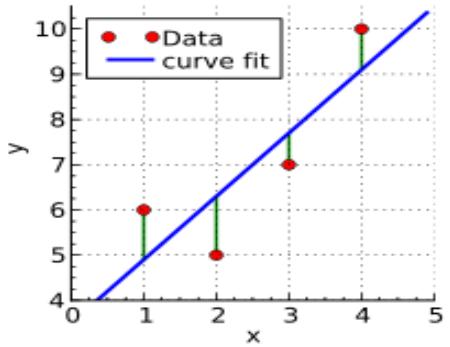
Update parameters for every two samples, batch size=2. Iterations = 12/2=6 per epoch

- Epoch
  - An epoch is when an entire train dataset is passed forward and backward through the neural network only once.
- Batch sizes
  - A total number of training examples used to update the model's parameters one time. During the training process, the model's parameters will be updated multiple times.
  - A reasonable ,number neither too small nor too large. A typical value is 2^n, n=3, 4, 5, 6.
- Iterations:
  - Number of batches need to complete one epoch

```
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-2.7959 -0.8575  0.0796  0.7418  2.4609

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.321      0.230   -1.40     0.17
x              2.473      0.407    6.08  2.3e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.12 on 98 degrees of freedom
Multiple R-squared:  0.274, Adjusted R-squared:  0.266
F-statistic: 36.9 on 1 and 98 DF,  p-value: 2.34e-08
```

How does lm figure out beta values?

Least Squares Estimation/ Ordinary least squares



4.3 Model training

- Loss function (minimize) / Objective function (optimize)
  - Quantitative: Mean squared error (least squares)

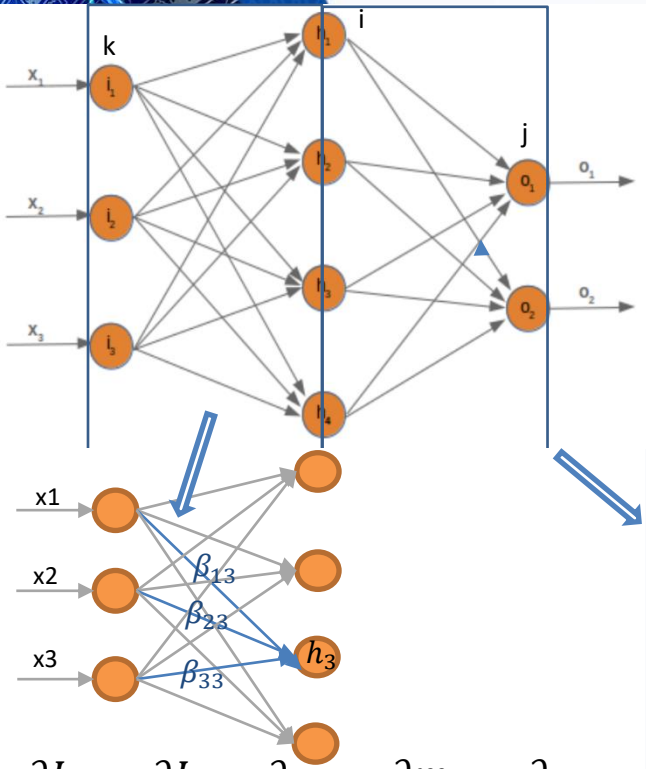
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

  - Binary or Categorical: cross-entropy

$$\text{Loss} = -\frac{1}{\substack{\text{output}\\\text{size}}}\sum_{i=1}^{\substack{\text{output}\\\text{size}}} y_i \cdot \log \hat{y}_i + (1-y_i)\cdot \log\,(1-\hat{y}_i)$$

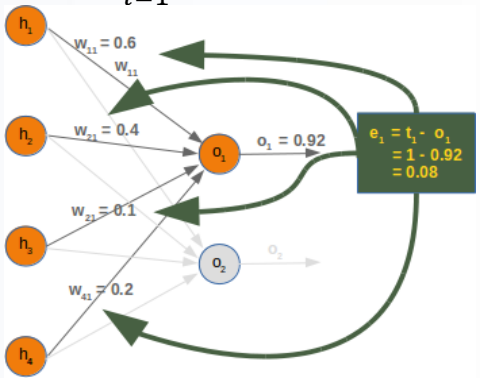
https://en.wikipedia.org/wiki/Linear_least_squares;

4.3 Model training



- Feed forward
  - input training data at the input layer, and it travels from input to hidden and from hidden to the output layer
- Back propagation
  - a widely used algorithm for training feedforward artificial neural networks
  - feed forward the values
  - calculate the error and propagate it back to the earlier layers.

$$m_j = \sum_{i=1}^{p} w_{ij} h_i + b_{ij} \quad o_j = \sigma(m_j) \quad L = \frac{1}{2}\sum_{j=1}^{n}(t_j - o_j)^2 \quad \frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial o_j} \times \frac{\partial o_j}{\partial w_{ij}}$$



$$Activation\ function: \sigma(x) = \frac{1}{1 + e^x}$$

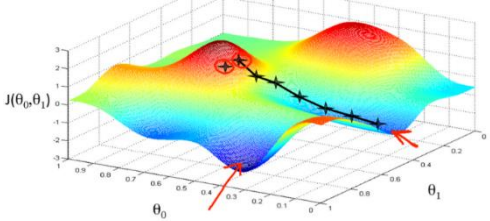
$$\frac{\partial L}{\partial w_{ij}} = \frac{\partial L}{\partial o_j} \times \frac{\partial \sigma}{\partial m_j} \times \frac{\partial m_j}{\partial w_{ij}}$$

$$\frac{\partial L}{\partial \beta_{ij}} = \frac{\partial L}{\partial o_j} \times \frac{\partial \sigma}{\partial m_j} \times \frac{\partial m_j}{\partial o_{wij}} \times \frac{\partial w_{ij}}{\partial \beta_{ki}}$$

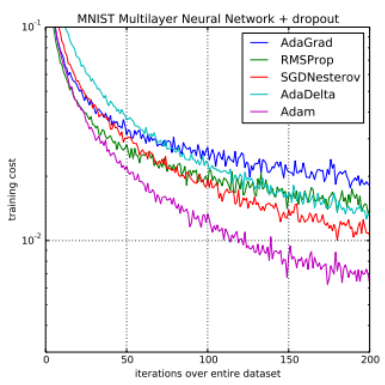
$$\frac{\partial L}{\partial w_{ij}} = (t_j - o_j) \times \left[\sigma(m_j)\left(1 - \sigma(m_j)\right)\right] \times h_1$$

https://medium.com/@jorgesleonel/backpropagation-cc81e9c772fd

4.3 Model training



- Optimizer:
  - SGD: stochastic gradient descent
  - Adam: Adaptive Moment Estimation



- Learning rate
  - α, controls how quickly the model is adapted to the problem. Range from (0,1) but usually a small positive value (e.g., 0.001).

```
tf.keras.optimizers.experimental.SGD(
    learning_rate=0.01,
    momentum=0.0,
    nesterov=False,
    amsgrad=False,
    weight_decay=None,
    clipnorm=None,
    clipvalue=None,
    global_clipnorm=None,
    use_ema=False,
    ema_momentum=0.99,
    ema_overwrite_frequency=None,
    jit_compile=True,
    name='SGD',
    **kwargs
)
```

```
tf.keras.optimizers.Adam(
    learning_rate=0.001,
    beta_1=0.9,
    beta_2=0.999,
    epsilon=1e-07,
    amsgrad=False,
    weight_decay=None,
    clipnorm=None,
    clipvalue=None,
    global_clipnorm=None,
    use_ema=False,
    ema_momentum=0.99,
    ema_overwrite_frequency=None,
    jit_compile=True,
    name='Adam',
    **kwargs
)
```

Gradient Descent:

$$\theta_i = \theta_i - \alpha \underbrace{\frac{\partial L}{\partial \theta_i}}_{\text{gradient}}$$

Momentum:

$$v_i = \gamma v_i + \alpha \frac{\partial L}{\partial \theta_i}$$

$$\theta_i = \theta_i - v_i$$

RMSProp:

$$v_i = \beta v_i + (1-\beta)\left(\frac{\partial L}{\partial \theta_i}\right)^2$$

$$\theta_i = \theta_i - \alpha \frac{\left(\frac{\partial L}{\partial \theta_i}\right)}{\sqrt{v_i} + \epsilon}$$

Adam:

$$m_i = \beta_1 m_i + (1-\beta_1)\frac{\partial L}{\partial \theta_i}$$

$$v_i = \beta_2 v_i + (1-\beta)\left(\frac{\partial L}{\partial \theta_i}\right)^2$$

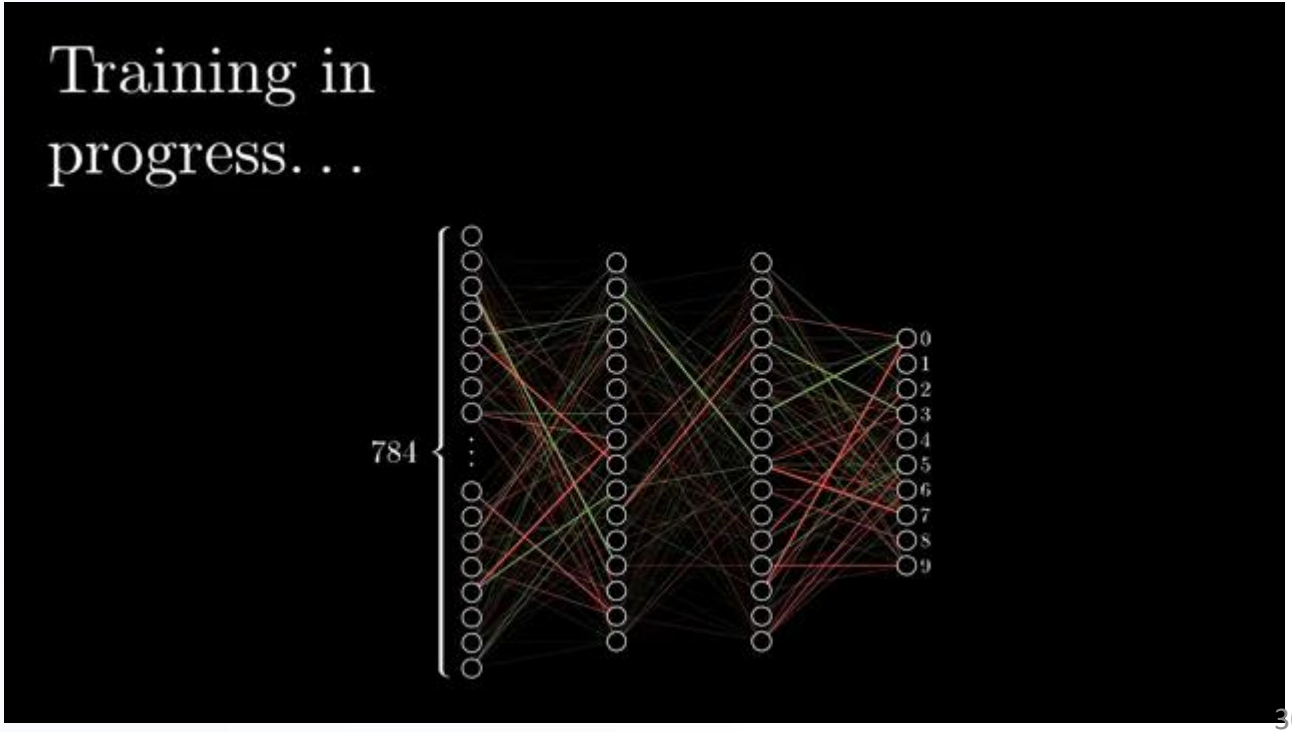
$$\widehat{m_i} = \frac{m_i}{1-\beta_1}, \qquad \widehat{v_i} = \frac{v_i}{1-\beta_2}$$

$$\theta_i = \theta_i - \frac{\alpha}{\sqrt{\widehat{v_i}} + \epsilon}\widehat{m_i}$$

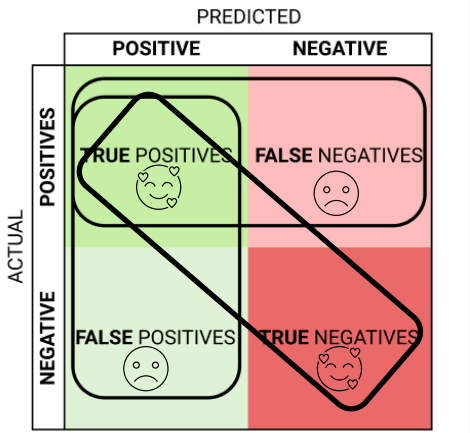
https://medium.com/mdr-inc/from-sgd-to-adam-c9fce513c4bb; https://medium.com/geekculture/a-2021-guide-to-improving-cnns-optimizers-adam-vs-sgd-495848ac6008

4.3 Model training

Model training in a nutshell:



https://gfycat.com/miniaturedependentcob



Precision

$$\frac{\textbf{TRUE POSITIVES}}{\textbf{TRUE POSITIVES} + \textbf{FALSE POSITIVES}}$$

Recall / TPR

$$\frac{\textbf{TRUE POSITIVES}}{\textbf{TRUE POSITIVES} + \textbf{FALSE NEGATIVES}}$$

F1 score

$$F_1 = 2 * \frac{precision * recall}{precision + recall} = \frac{tp}{tp + \frac{1}{2}fp + fn}$$

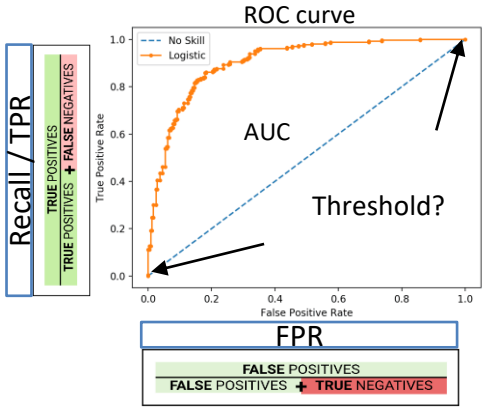
Accuracy

$$\frac{TP + TN}{TP + TN + FP + FN}$$

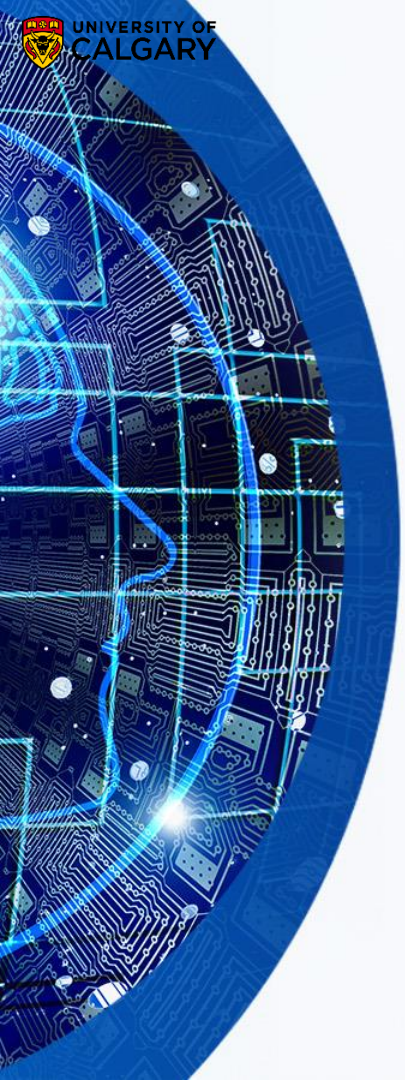
4.4 Model evaluation

- Metrics:
  - Quantitative outputs: Pearson R, Coefficient of determination R^2,
  - Dichotomized outputs: accuracy, precision, recall, F1 score, confusion matrix, and AUC (area under the ROC curve)

1: dog, 0: cat	Threshold	0	0.2	0.4	0.6	0.8
Sample 1 (O1=1, O2=0)	O1 = 0.9 O2 = 0.1	1, 0	1, 0	1, 0	1, 0	1, 0
Sample 2 (O1=0, O2=1)	O1 = 0.7 O2 = 0.3	1, 0	1, 0	1, 0	1, 0	0, 1
Sample 3 (O1=0, O2=1)	O1 = 0.5 O2 = 0.5	1, 0	1, 0	1, 0	0, 1	0, 1
Sample 4 (O1=1, O2=0)	O1 = 0.3 O2 = 0.7	1, 0	1, 0	0, 1	0, 1	0, 1



https://towardsdatascience.com/precision-and-recall-88a3776c8007



4.4 Model evaluation

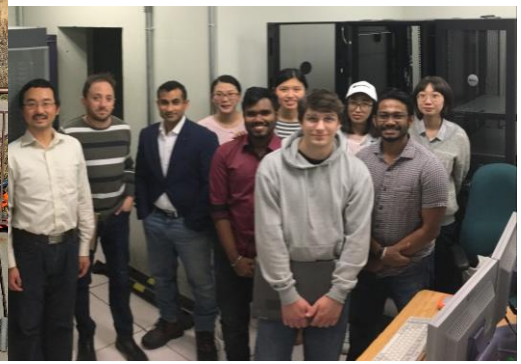
- Evaluation metrics:
  - Regression:
    - Pearson R, Coefficient of determination R^2,
  - Classification:
    - accuracy, precision, recall, F1 score, confusion matrix, and AUC. Which is the optimal one? Depends on specific user-cases.
  - General notes:
    - Precision: more confident of your true positives (e.g., spam emails)
    - Recall: false positives cost less than false negatives (aka false negatives are intolerable) (e.g., gas leakage)
    - F1 score: false positives and false negatives **not balance**
    - Accuracy: false positives and false negative **balances** and they have **similar costs**
    - AUC: Model's overall performance, not point performance at a certain threshold
- Evaluate the model on the validation dataset to optimize hyperparameters, stop earlier, measure model's performance and select optimal models.

5. Hands-on: Image classification

- Task: Transfer pre-trained model MobileNet V2 on the classification of images of dogs and cats.
  - 1. Load and examine pre-trained MobileNetV2
  - 2. Add additional layers to form a new model
  - 3. Load and examine datasets
  - 4. Train the new model
  - 5. Fine-tune the new model
  - https://colab.research.google.com/drive/1p7lmhIh1AUKJx-aZnbdsBoqmpS3p0DrS

What is Colab? https://colab.research.google.com/
Colab, or 'Colaboratory', allows you to write and execute Python in your browser, with
•Zero configuration required
•Access to GPUs free of charge
•Easy sharing

Acknowledgment



BMB
DEPARTMENT OF BIOCHEMISTRY AND MOLECULAR BIOLOGY

https://sites.google.com/site/quanlongresearch/?pli=1

UNIVERSITY OF CALGARY

Thank you