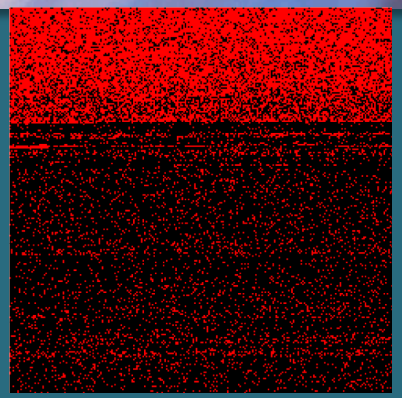
Frederick National Laboratory
for Cancer Research
sponsored by the National Cancer Institute

Cancer Type/Site Classification using Deep-Learning

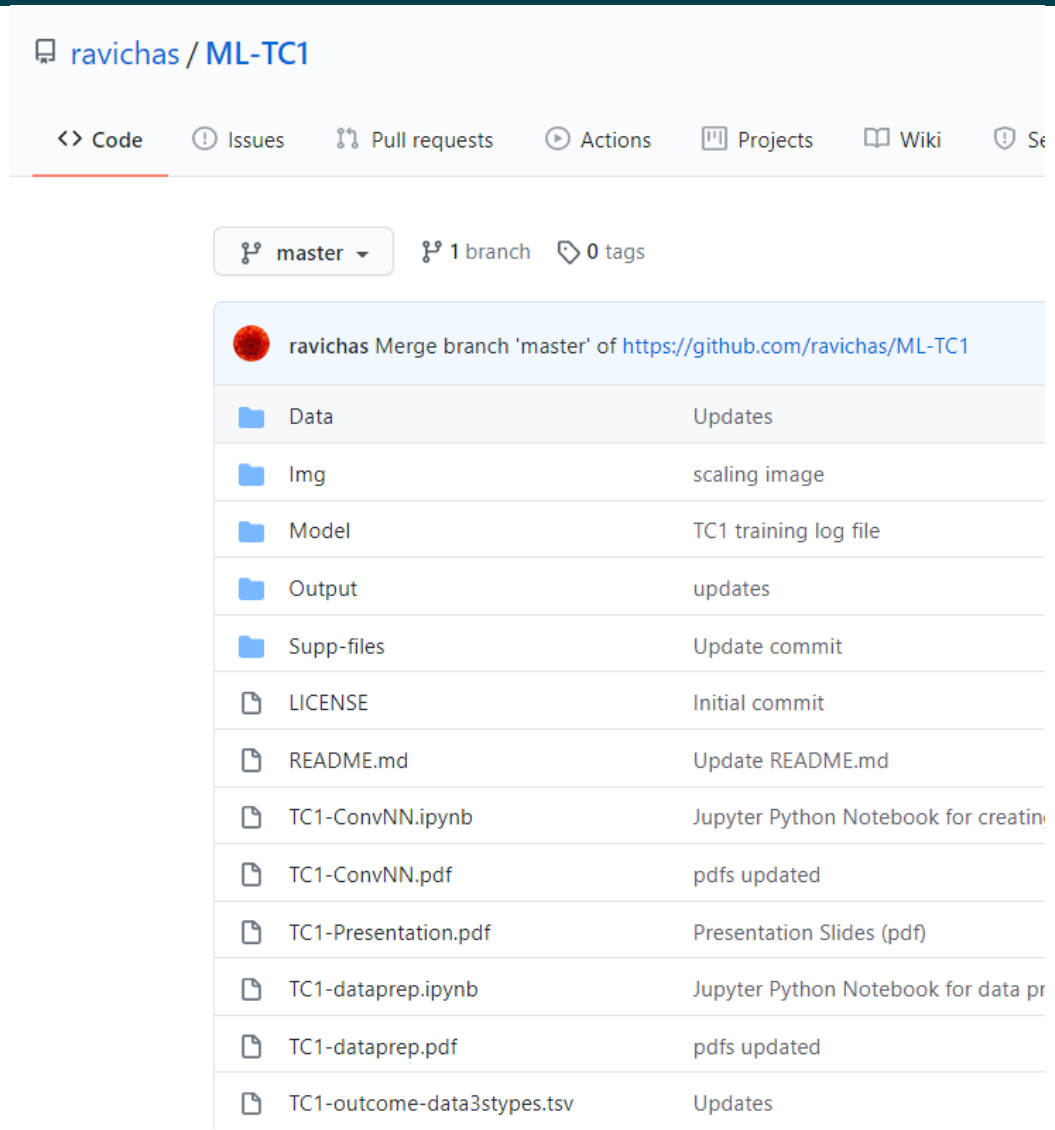
S. Ravichandran, Ph.D
BIDS, FNLCR



DEPARTMENT OF HEALTH AND HUMAN SERVICES • National Institutes of Health • National Cancer Institute
Frederick National Laboratory is a Federally Funded Research and Development Center operated by Leidos Biomedical Research, Inc., for the National Cancer Institute

Supporting link: https://github.com/ravichas/ML-TC1

Frederick National Laboratory for Cancer Research
sponsored by the National Cancer Institute



1. **TC1-Presentation.pdf**
   PPT slides in PDF

2. **TC1-dataprep.pdf** and **TC1-ConvNN.pdf** are the pdf versions of the **TC1-dataprep.ipynb** and **TC1-ConvNN.ipynb** Jupyter Notebooks

3. **Data** folder contains the data files

4. **Model** folder will contain Model related weights

Biowulf HPC Batch Job scripts

/data/BIDS-HPC/public/Workshops/Ravi/ML-TC1.tar.gz

Contents of *tar.gz file:

- Scripts & README.txt
- Python code
  SLURM script
- Data

Check the README.txt file for some preliminary setup

Acknowledgements

- **NCI-DOE Pilot-1 Team** especially Maulik Shukla

- **FNL/BIDS**
  - Drs. Andrew Weissman, Mark Jensen George Zaki and Eric Stahlberg
  - Dr. Deb Hope, Anney Che, Hue Reardon, Naomi Ohashi, Dr. Yongmei Zhao
  - Amar Khalsa, Laurie Morrissey, Petrina Hollingsworth and Lynn Borkon
  - Colleagues who reviewed the material

https://datascience.cancer.gov/collaborations/joint-design-advanced-computing

The Joint Design of Advanced Computing Solutions for Cancer (JDACS4C)

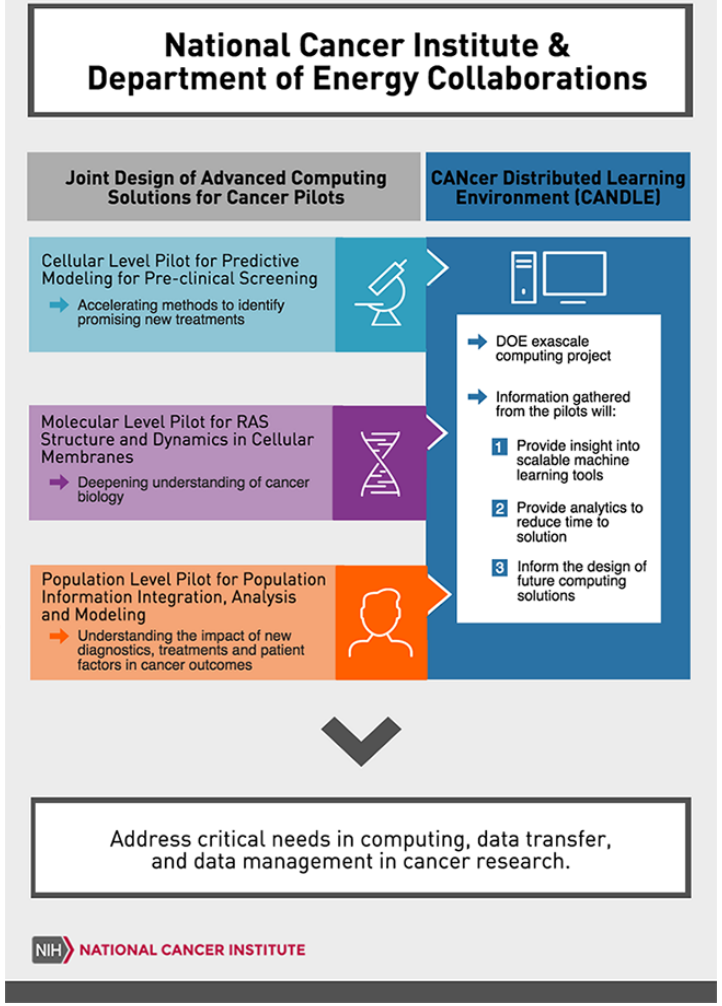
Frederick National Laboratory for Cancer Research
sponsored by the National Cancer Institute

- JDACS4C program was created in 2016 to accelerate cancer research using emerging exascale computing capabilities.
- Part of the Cancer Moonshot
- Cross-agency collaboration between NCI and the DOE
- **Pilot1:**
  - *Focuses on developing predictive models, both computational and experimental, to improve pre-clinical therapeutic drug screening.*
  - https://datascience.cancer.gov/collaborations/joint-design-advanced-computing/cellular-pilot

Pilot1

Pilot2

Pilot3



Introduction

- **Goal is to share tools/techniques/solutions for cancer related problems**
- **You would be able to take our test-case (code/scripts) and tune it to your needs**
- **Deep-Learning is a growing area. This project may not address all your questions, but I believe this will be a good starting point**
- **We want to hear from you, please send us your feed-back**

Motivation: Cancer Prediction vs Cancer Detection

- **Cancer Prediction has been the major focus**
  - Prognosis, Recurrence, Susceptibility

- **Cancer Detection (classification of tumors/cancers) is lagging behind Prediction and we would like to share an application that might be useful**
  - Detect/Identify cancer type at an early stage

Goal(s)/Questions

- **Take genomic expression data from tumor/cancer samples and apply Deep-Learning to create cancer types/site(s) classifier models**

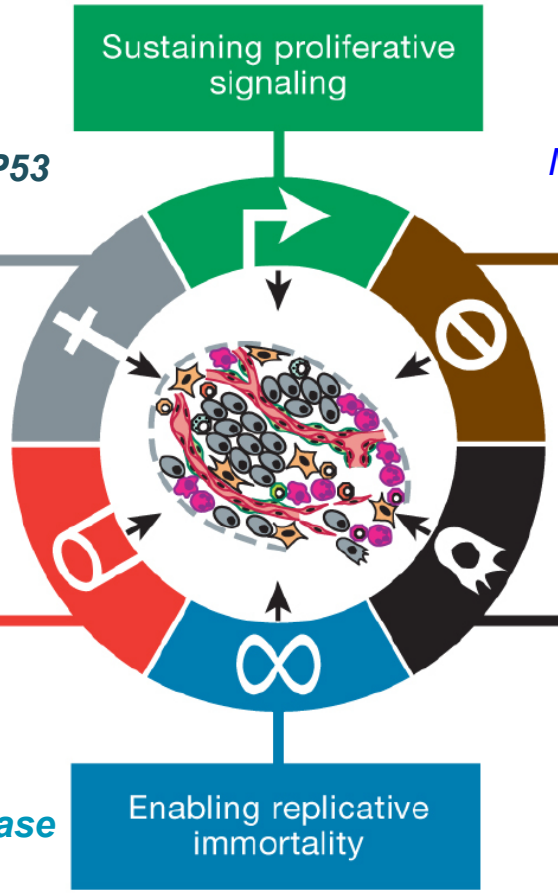
- **Are the expression profiles unique to be used for early cancer detection?**
  - Improving chance of early detection cure/survival?

Hallmarks of cancer: Integral Components of Most Forms of Cancer (Acquired Capabilities)

Frederick National Laboratory for Cancer Research
sponsored by the National Cancer Institute

Hallmarks of Cancer: The Next Generation

src Oncogene mutations



Mutation or missing CASP9 or P53

Mutations in P53 tumor suppressor

New blood vessels via VEGF

Mutations in cell-adhesion CDH1

Mutations in telomerase

Hanahan and Weinberg, 2011

Cell PRESS

REVIEW | VOLUME 100, ISSUE 1, P57-70, JANUARY 07, 2000

The Hallmarks of Cancer

Douglas Hanahan • Robert A Weinberg

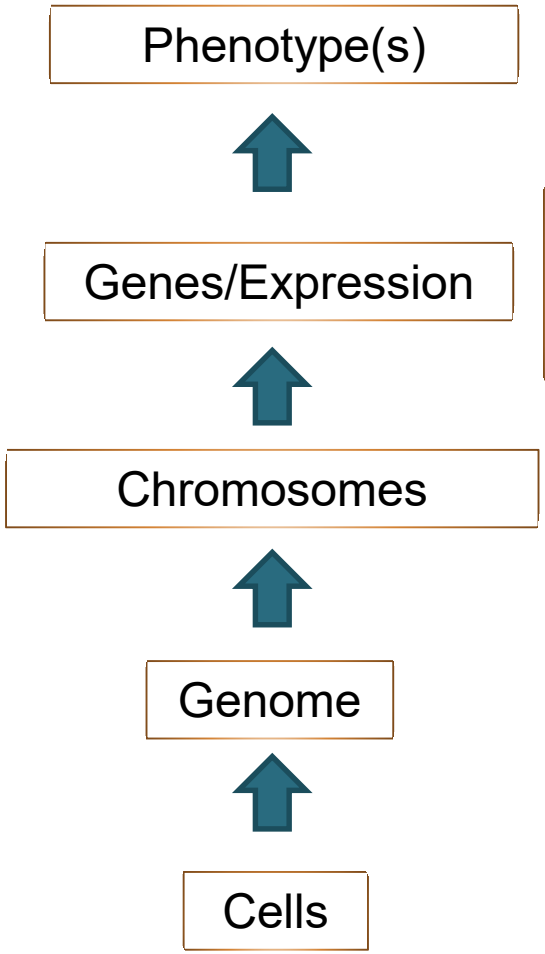
Open Archive • DOI: https://doi.org/10.1016/S0092-8674(00)81683-9

Overview of Genotype/phenotypes?

Influence of genomic features on phenotypes: An overview

Lung-cancer

RNASeq profile



"The observable characteristics in an individual resulting from the expression of genes" from NCI

Promoters
Enhancers
Silencers

Epigenetic factors
DNA methylation
Histone modifications

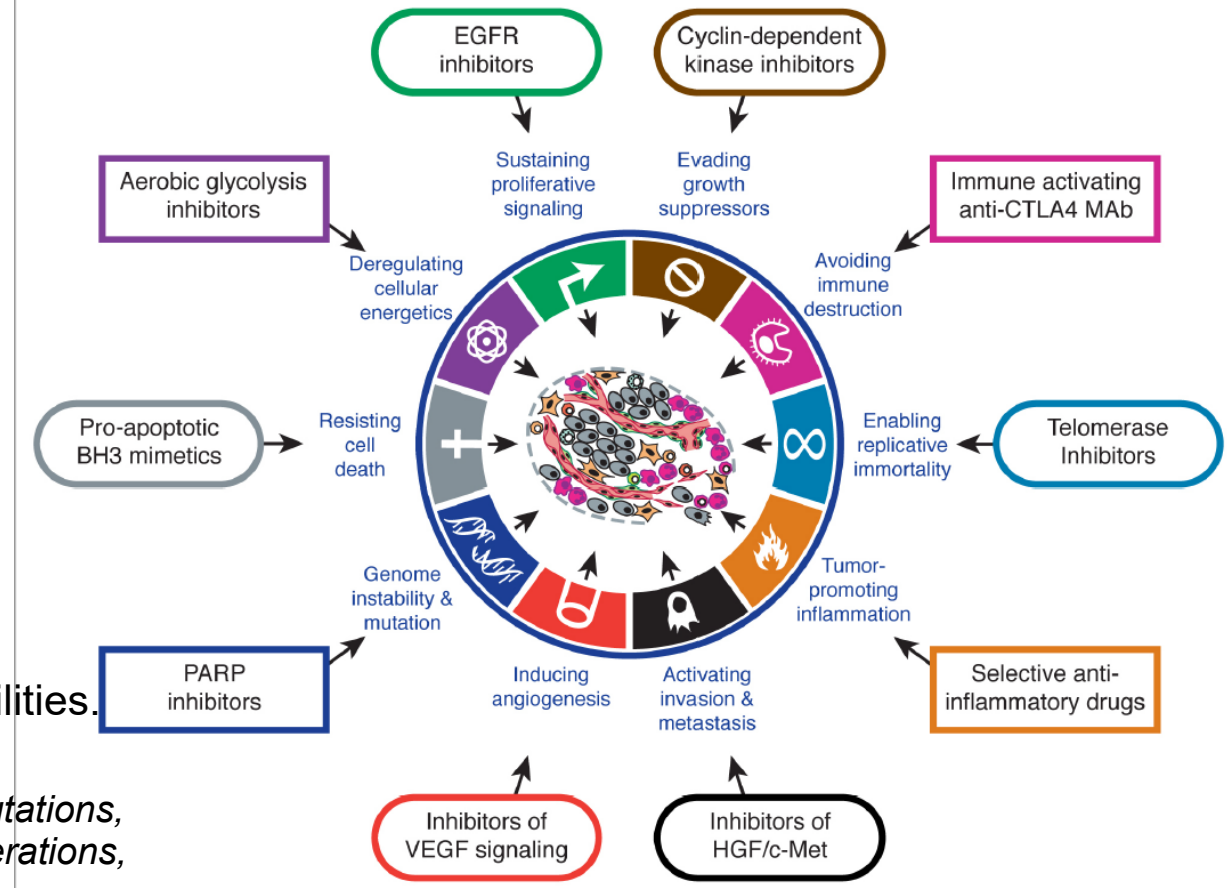
Variations

Diagnosis/treatment vs Prediction

Treatment vs Type-Prediction

- **Treatment**
  - Gene-centric (or a slice of pathway)
  - Disease:
    - Tumor is called a gastrointestinal stromal tumor, or GIST
    - Medicine/inhibitor: Imatinib targeting BCR/KIT

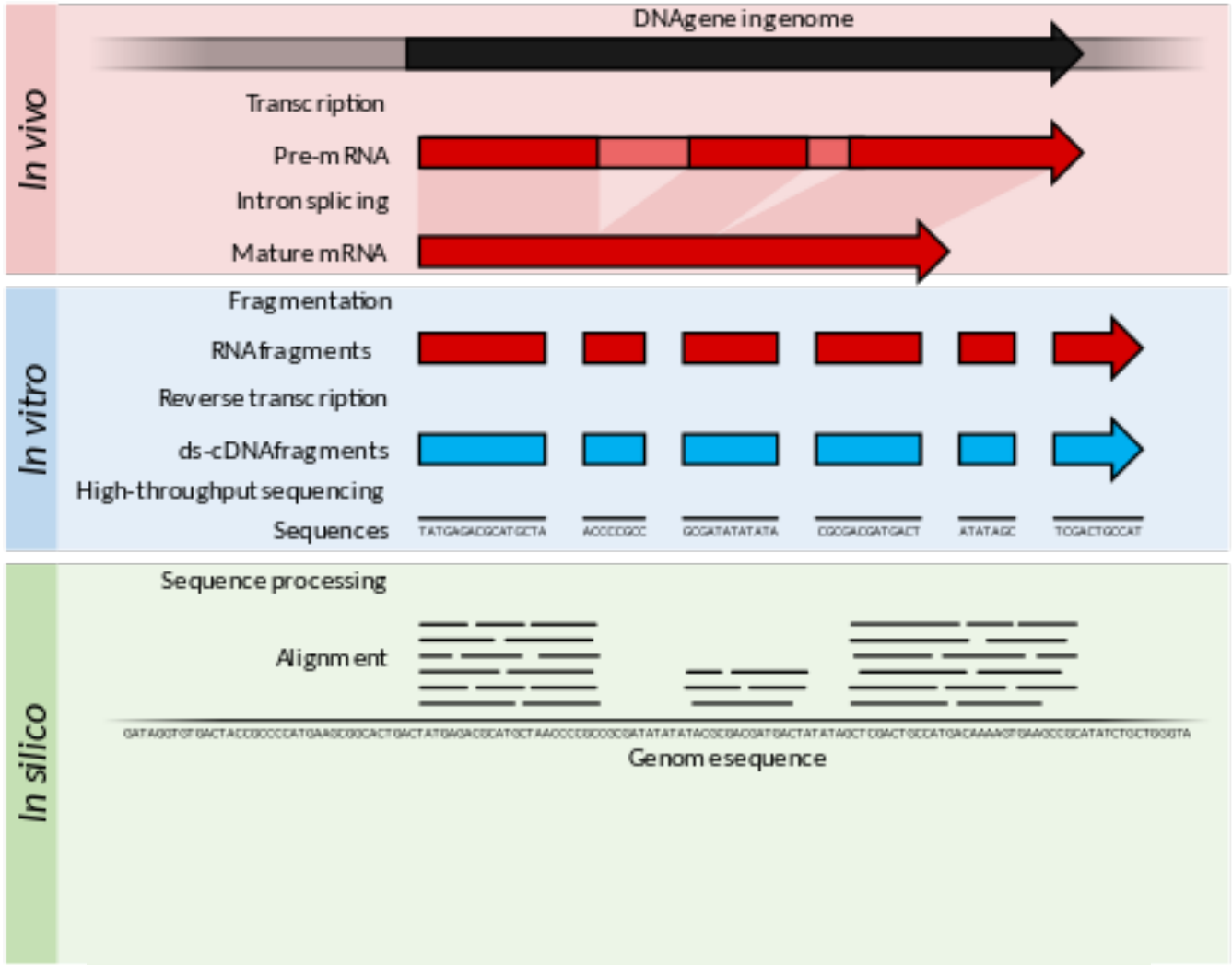
- **Detecting Type**
  - Genomic instability in Cancer Cells → Random mutations → rare genetic changes that can orchestrate hallmark capabilities. *(Hanahan and Weinberg 2011)*
  - *"The architecture of occurring genetic aberrations such as somatic mutations, CNVs, changed gene expression profiles, and different epigenetic alterations, is unique for each type of cancer.",*
    DOI: 10.5114/wo.2014.47136
  - https://pubmed.ncbi.nlm.nih.gov/26963104/ (PLOS, 2016)



Hanahan and Weinberg, 2011

Cell PRESS

Expression data



NGS

Spliced to become mature mRNA

mRNA is extracted

mRNA captured/fragmented/copied into stable ds-cDNA

Sequenced

NGS

Reference Genome

Data source: The Cancer Genome Atlas (TCGA)

- NIH launched TCGA Pilot Project – a public funded project
- Goal of creating a comprehensive “atlas” of cancer genomic profiles.
- Large cohorts of over 30 human tumors through large-scale genome sequencing and integrated multi-dimensional analyses.
- Contains Microarray and NGS data
  - RNASeq
  - miRNA seq
  - SNP based platforms
  - …..
- TCGA data is available via GDC

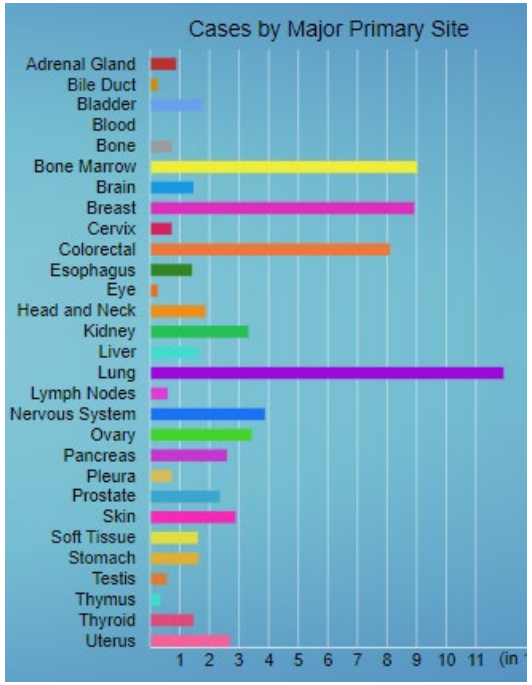
https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga

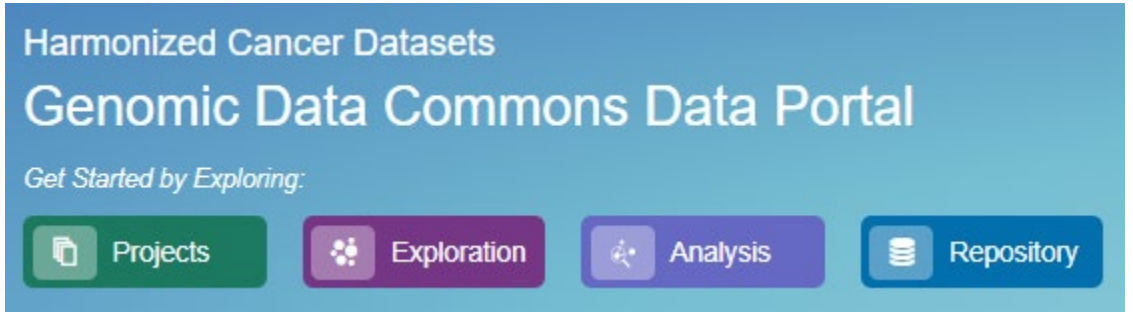
Data Harmonization: GDC (https://gdc.cancer.gov/)

- **Data and metadata is submitted to the GDC in standard data types and file formats. Other data sources (Ex. TCGA) are also included**

- **Data are harmonized against a common reference genome (GRCh38)**

- **For this workshop, we will focus on TCGA Genomic expression data from GDC**

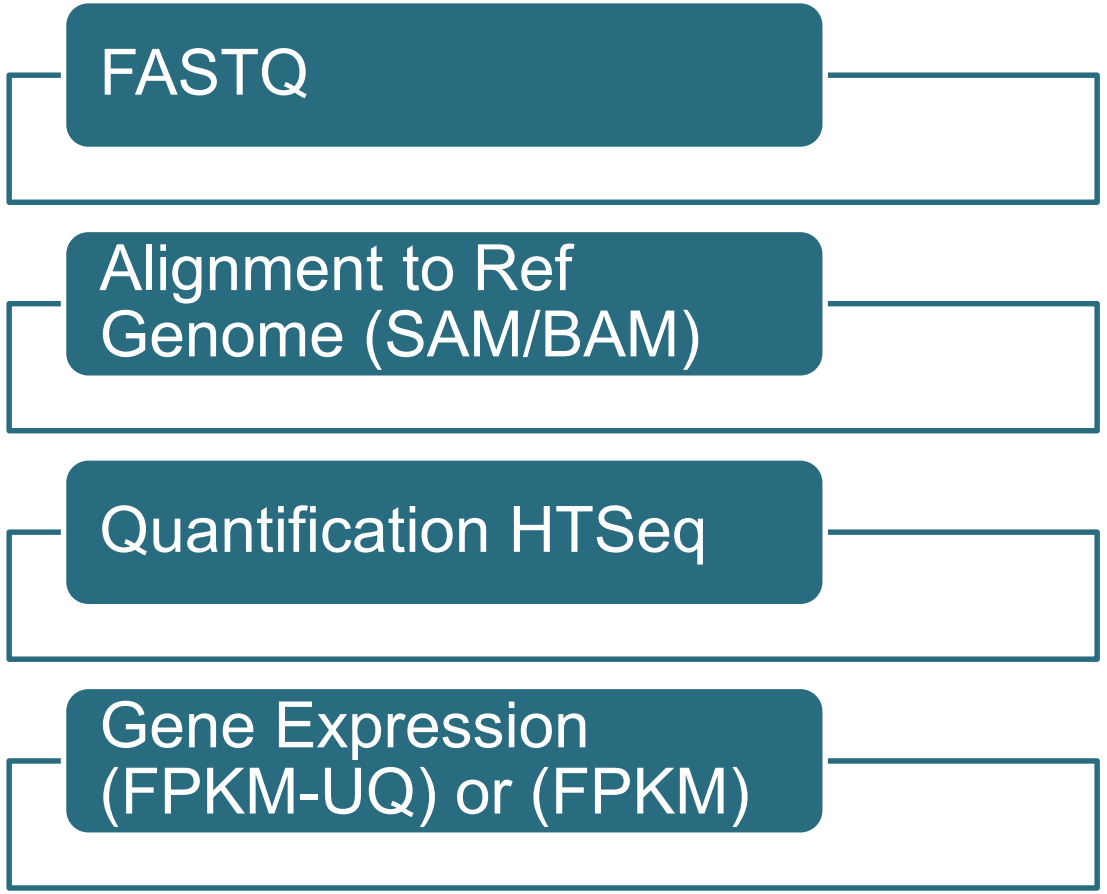




Expression Data Quantification

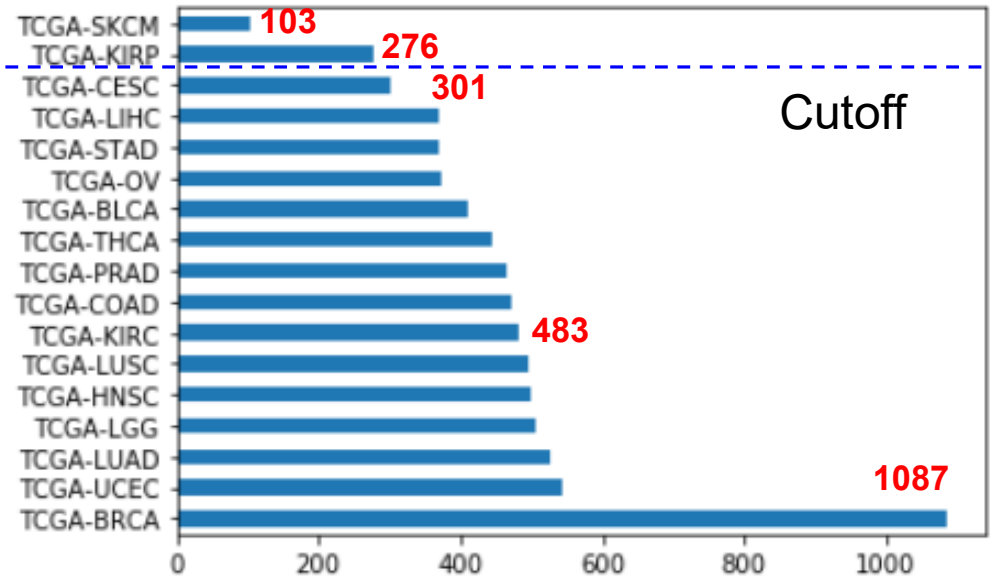
- RC_g: Number of reads mapped to the gene
- RC_{g75}: The 75th percentile read count value for genes in the sample
- L: Length of the gene in base pairs; Calculated as the sum of all exons in a gene

$$\text{FPKM-UQ} = \frac{\text{RC}_g \times 10^9}{\text{RC}_{g75} \times \text{L}}$$



Fragments Per Kilobase of transcript per Million mapped reads

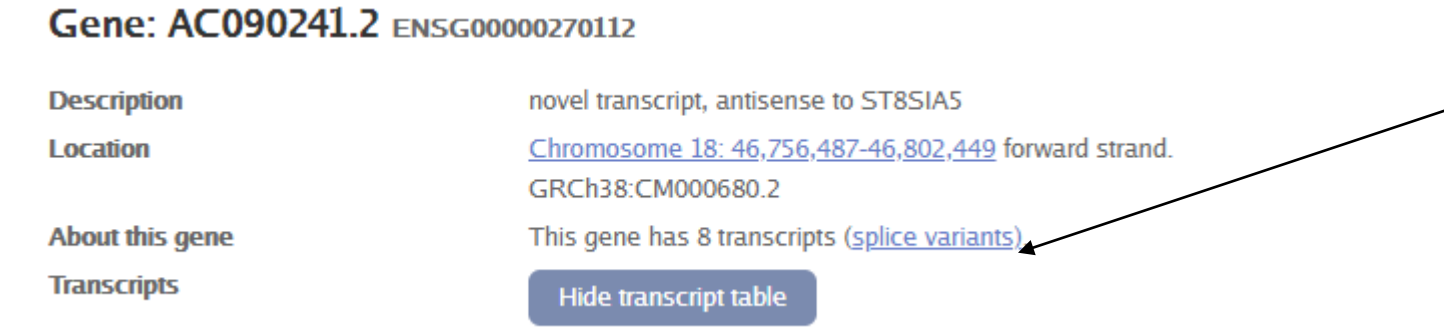
How much data for modeling?

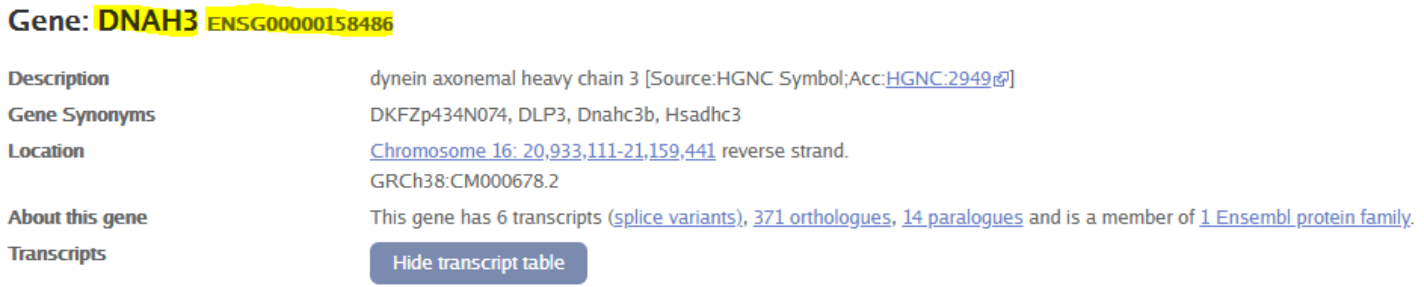


CODE	Cancer Site/Type
BRCA	Breast invasive carcinoma
UCEC	Uterine Corpus Endometrial Carcinoma
LUAD	Lung adenocarcinoma
LGG	Brain Lower Grade Glioma
HNSC	Head and Neck squamous cell carcinoma
LUHSC	Lung squamous cell carcinoma
KIRC	Kidney renal clear cell carcinoma
PRAD	Prostate adenocarcinoma
COAD	Colon adenocarcinoma
THCA	Thyroid carcinoma
BLCA	Bladder Urothelial Carcinoma
OV	Ovarian serous cystadenocarcinoma
STAD	Stomach adenocarcinoma
LIHC	Liver hepatocellular carcinoma
CEC	Cervical squamous cell carcinoma and endocervical adenocarcinoma

300 samples each

Expression data from a sample





TCGA-BRCA

Genes	Expression
ENSG00000242268.2	1658.464179
ENSG00000270112.3	460.2343433
ENSG00000167578.15	52440.10096
ENSG00000273842.1	0
ENSG00000078237.5	68165.45626
ENSG00000146083.10	255959.2351
ENSG00000225275.4	0
ENSG00000158486.12	104.9473768
ENSG00000198242.12	4968556.658
ENSG00000259883.1	6108.999052
ENSG00000231981.3	0
ENSG00000269475.2	0
ENSG00000201788.1	0
ENSG00000134108.11	957330.2056
ENSG00000263089.1	3484.027373
ENSG00000172137.17	41485.9507
ENSG00000167700.7	226717.4208
ENSG00000234943.2	2082.245035
ENSG00000240423.1	310.5246749
ENSG00000060642.9	155863.9216
ENSG00000271616.1	0
ENSG00000234881.1	0
ENSG00000236040.1	394.4755669
ENSG00000231105.1	1583.312582
ENSG00000243044.1	0
ENSG00000182141.8	45538.60648
ENSG00000269416.4	119.0847054
ENSG00000264981.1	0

60,483 transcripts

Data Preparation

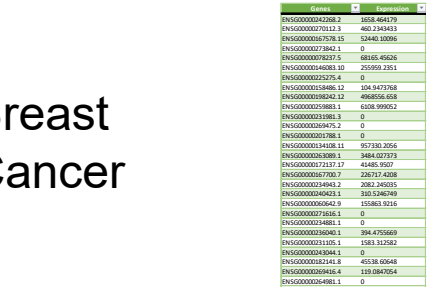
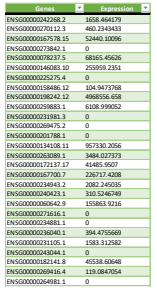
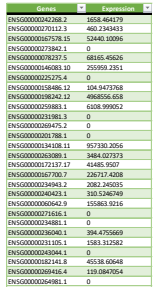
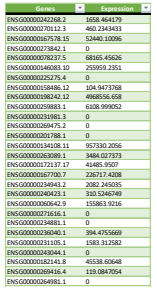
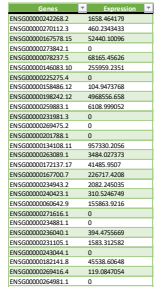
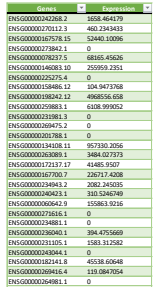
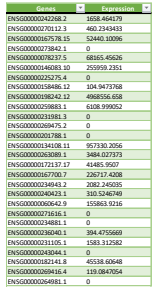
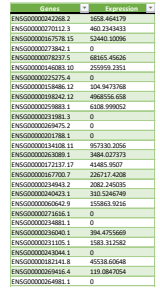
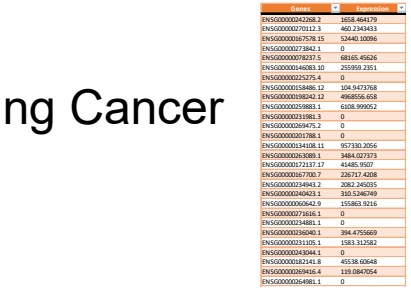
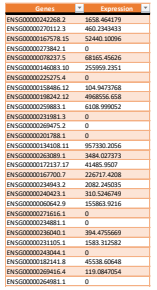
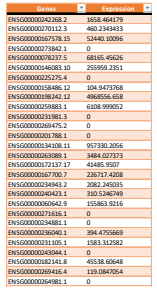
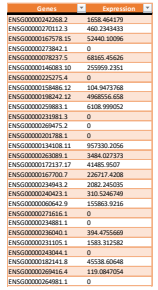
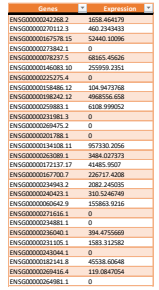
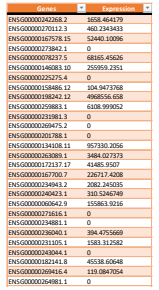
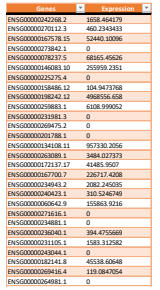
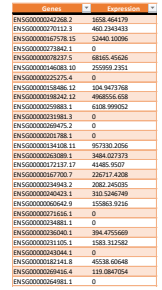
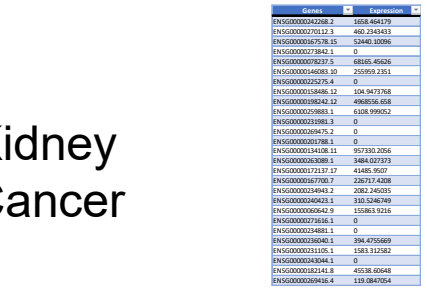
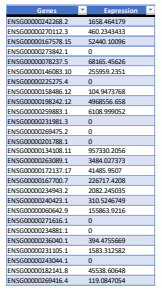
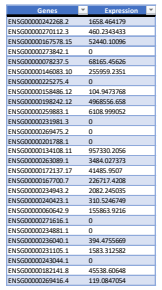
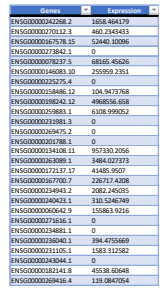
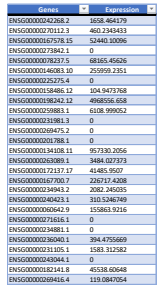
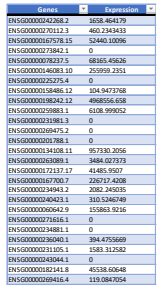
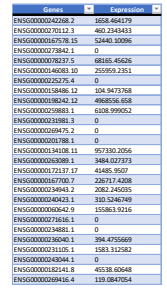
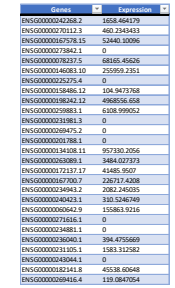
Breast Cancer

60,484 transcripts

	Sample1	Sample2	Sample3	Sample4		Sample297	Sample298	Sample299	Sample300
Gene	Expression								
ENSG00000242268.2	1658.464179								
ENSG00000270112.3	460.2343433								
ENSG00000167578.15	52440.10096								
ENSG00000273842.1	0								
ENSG00000078237.5	68165.45626								
ENSG00000146083.10	255959.2351								
ENSG00000225275.4	0								
ENSG00000159486.12	104.9473768								
ENSG00000198242.12	4968556.658								
ENSG00000259883.1	6108.999052								
ENSG00000231981.3	0								
ENSG00000269475.2	0								
ENSG00000201788.1	0								
ENSG00000134108.11	957330.2056								
ENSG00000263089.1	3484.027373								
ENSG00000172137.17	41485.9507								
ENSG00000167700.7	226717.4208								
ENSG00000234943.2	2082.245035								
ENSG00000240423.1	310.5246749								
ENSG00000060642.9	155863.9216								
ENSG00000271616.1	0								
ENSG00000234881.1	0								
ENSG00000236040.1	394.4755669								
ENSG00000231105.1	1583.312582								
ENSG00000243044.1	0								
ENSG00000182141.8	45538.60648								
ENSG00000269416.4	119.0847054								
ENSG00000264981.1	0								

Data Preparation

Frederick National Laboratory for Cancer Research
sponsored by the National Cancer Institute

	Sample1	Sample2	Sample3	Sample4	.....	Sample297	Sample298	Sample299	Sample300
Breast Cancer									
Lung Cancer									
Kidney Cancer									

Merged Sample Expression Data

Genes

SAMPLES

	0	1	2	3	4	5	6	7	8	9	...	60474	60475	60476	60477	60478	60479	60480	60481	60482	submitter_id
0	574548	2263.14	983212	69718	54834.9	19718.1	175853	735123	38662.4	233190	...	0	0	0	0	0	0	0	0	0	TCGA-04-1331-01A-01R-1569-13
1	352295	4592.37	663107	39745.4	36553.5	41147.1	241313	396423	37567	128693	...	0	0	0	0	0	0	0	0	0	TCGA-04-1332-01A-01R-1564-13
2	295162	649.026	1.21115e+06	57385.5	33097.4	58051.8	228615	346066	105567	408267	...	0	0	0	0	0	0	0	0	0	TCGA-04-1338-01A-01R-1564-13
3	329580	1835.59	1.08437e+06	33812.3	24516.1	22330.6	42134.4	895558	56178	83847.3	...	0	0	0	0	0	0	0	0	0	TCGA-04-1341-01A-01R-1564-13
4	289269	40061.7	2.44837e+06	26399.5	18248	49610	74761.1	571992	71951.9	98726.4	...	0	0	0	0	0	0	0	0	0	TCGA-04-1343-01A-01R-1564-13
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4495	1.18093e+06	0	1.01139e+06	67877.2	15005.7	50527.3	6.21536e+06	1.47373e+06	459656	167488	...	0	0	0	0	0	0	0	0	0	TCGA-ZS-A9CD-01A-11R-A37K-07
4496	929228	0	869800	95607.5	17188.6	9352.12	7.61121e+06	196838	354465	138074	...	0	0	0	0	0	0	0	0	0	TCGA-ZS-A9CE-01A-11R-A37K-07
4497	469276	476.683	516938	110051	34469.4	37334.7	5.95811e+06	427832	323833	154861	...	0	0	0	0	0	0	0	0	0	TCGA-ZS-A9CF-01A-11R-A38B-07
4498	2.44119e+06	18282.7	853547	79288.7	106926	42593.9	4.80111e+06	955338	331924	177020	...	0	0	0	0	0	0	0	0	0	TCGA-ZS-A9CG-01A-11R-A37K-07
4499	259853	505.488	591328	74253.7	42553.5	118772	148978	508465	153862	170412	...	0	0	0	0	0	0	0	0	0	TCGA-ZX-AA5X-01A-11R-A42T-07

4500 rows × 60484 columns

Transpose and add as a row

Quantifying mRNA abundance and Scaling

- **Use GDC harmonization expression data (X = FPKM or FPKM-UQ)**

- **FPKM-UQ or FPKM is rescaled to TPM using the following formula.**

Thanks to Andrew for his help in simplifying the scaling slides

$$\mathrm{TPM}_i = \left(\frac{\mathrm{X}_i}{\Sigma_j \mathrm{X}_j}\right) \cdot 10^6$$

- **TPM has nice mathematical properties and a stable entity and can be compared across samples**

https://docs.gdc.cancer.gov/Encyclopedia/pages/HTSeq-FPKM-UQ/

Mapping and quantifying mammalian transcriptomes by RNA-Seq

Ali Mortazavi[1,2], Brian A Williams[1,2], Kenneth McCue[1], Lorian Schaeffer[1] & Barbara Wold[1]

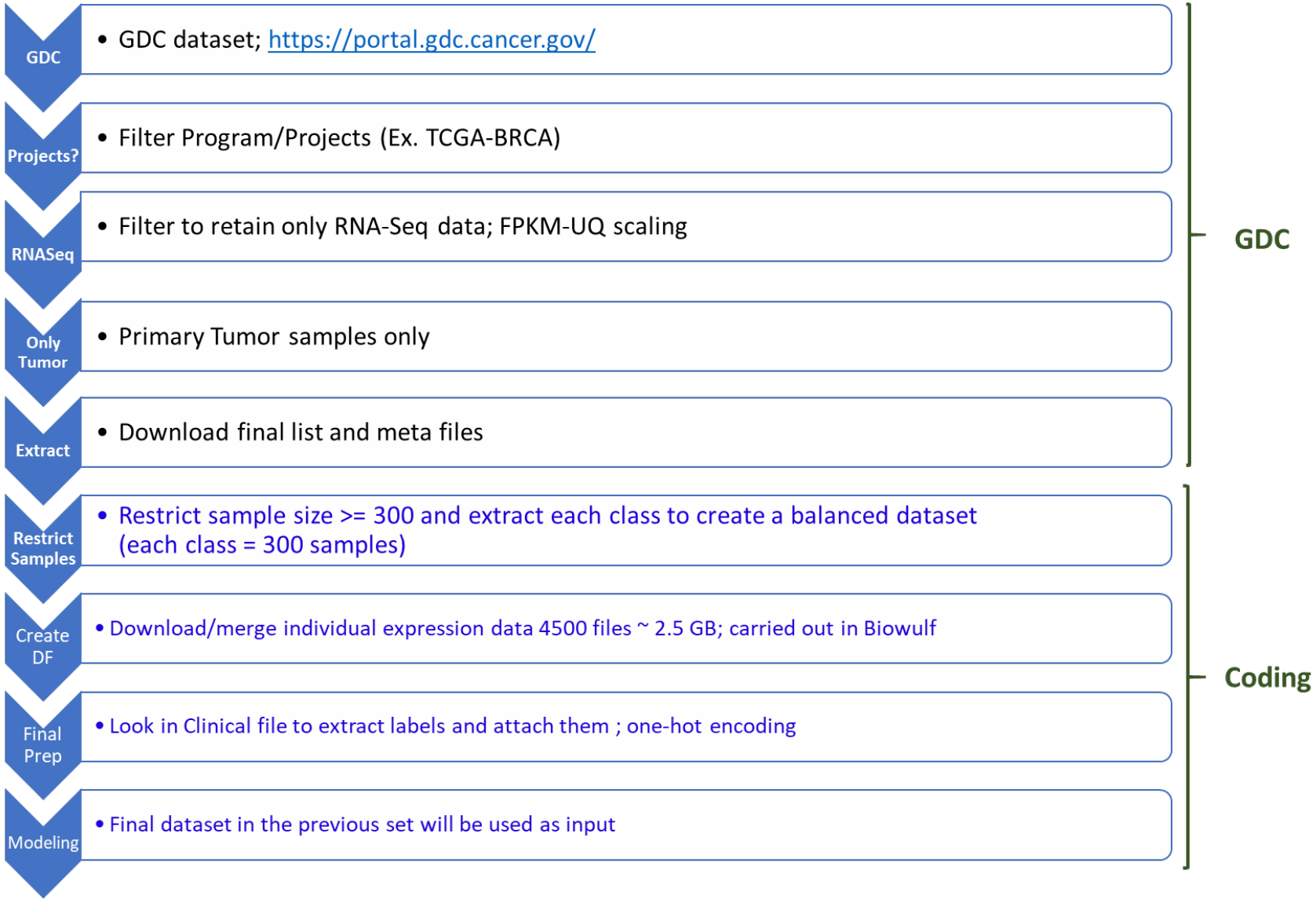
One-hot encoding to convert Cancer types to numbers

- **Convenient to transform categorical variables into a numerical quantity for computations**
  - BRCA to 0 ; LUAD to 1 etc.
  - 0, 1, 2, 3, …, 13, 14

TCGA-CESC
TCGA-LIHC
TCGA-STAD
TCGA-OV
TCGA-BLCA
TCGA-THCA
TCGA-PRAD
TCGA-COAD
TCGA-KIRC
TCGA-LUSC
TCGA-HNSC
TCGA-LGG
TCGA-LUAD
TCGA-UCEC
TCGA-BRCA

```
>>> encoded
array([[1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 1.]],
      dtype=float32)
```

Data preparation steps summary



Before we break for hands-on

- **Python as the programming language for this workshop, but similar libraries are available in R or other languages**

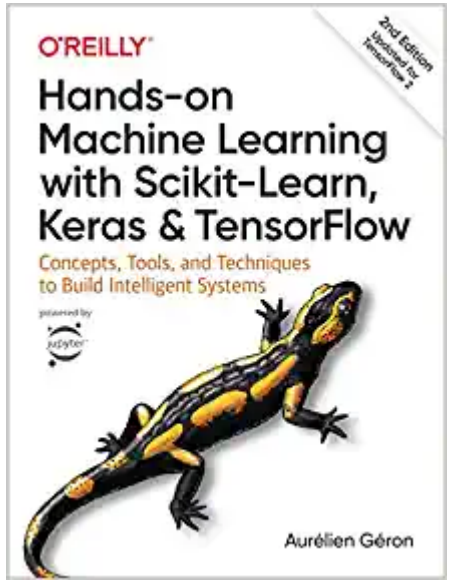
- **Will use Jupyter Notebook for sharing the code**
  - With little effort one can convert the Python code into R and still use Jupyter Notebook

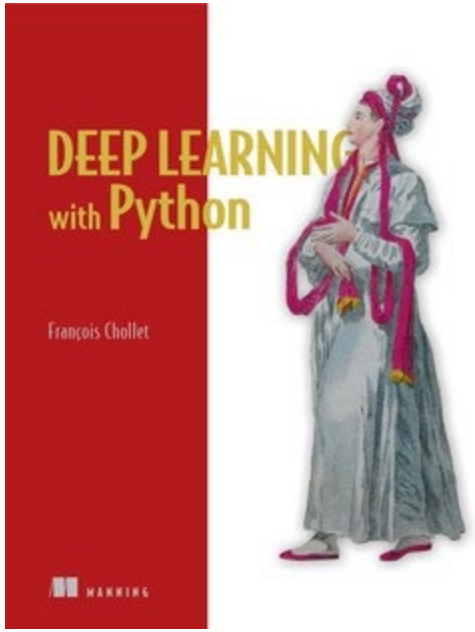
To be continued after hands-on

https://github.com/ravichas/ML-TC1

Before we begin the modeling section …

- **Due to lack of time, I wont be covering the basics of Neural Network**





These are good books for beginners and up

Keras is a high-level NN package that is built on top of popular high-level libraries (TF, Theano). Works well with CPU/GPU

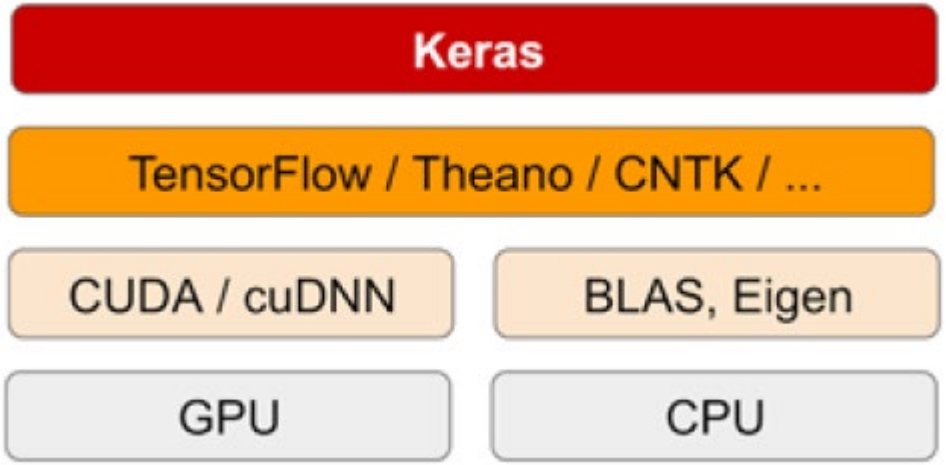


Figure from Deep Learning with Python

Supervised Learning

- Goal
  - Construct a model that takes in input features/target pair to return a prediction for target/outcome

- Train a machine learning
  - Model refers to learning its parameters (for an Architecture), which typically involves minimizing a loss function on training data with the aim of making accurate predictions on unseen (test) data

Supervised Learning:

Data: (x,y) ; where x is the genomic expression profile ; y is the cancer classes

Goal? Learn the function that maps
x → y

Terminology

	0	1	2	3	4	5	6	7	8	9	...	60474	60475	60476	60477	60478	60479	60480	60481	60482	submitter_id
0	574548	2263.14	983212	69718	54834.9	19718.1	175853	735123	38662.4	233190	...	0	0	0	0	0	0	0	0	0	TCGA-04-1331-01A-01R-1569-13
1	352295	4592.37	663107	39745.4	36553.5	41147.1	241313	396423	37567	128693	...	0	0	0	0	0	0	0	0	0	TCGA-04-1332-01A-01R-1564-13
2	295162	649.026	1.21115e+06	57385.5	33097.4	58051.8	228615	346066	105567	408267	...	0	0	0	0	0	0	0	0	0	TCGA-04-1338-01A-01R-1564-13
3	329580	1835.59	1.08437e+06	33812.3	24516.1	22330.6	42134.4	895558	56178	83847.3	...	0	0	0	0	0	0	0	0	0	TCGA-04-1341-01A-01R-1564-13
4	289269	40061.7	2.44837e+06	26399.5	18248	49610	74761.1	571992	71951.9	98726.4	...	0	0	0	0	0	0	0	0	0	TCGA-04-1343-01A-01R-1564-13

- **Columns**
  - input variables or features or attributes
- **Outcome column**
  - Outcome variables or targets
- **Rows**
  - Training example or instance
- **Whole table Training data set**

What is different about Neural Network?

- If you know the equation (algorithm), then you feed in the input and you get the output. You can code the function yourself

```
def function(x):
    y = 2.0 + 5.0 * x
    return(y)
```

- You can choose to use linear modeling and use the data to figure the relationship

```
Model ← lm( y ~ x)
```

- Neural Network using the data learn the algorithm.

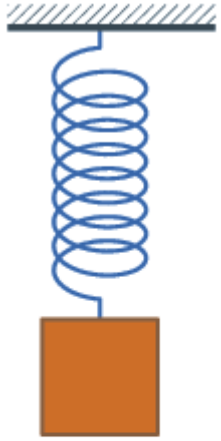
INPUT

ALGORITHM

OUTPUT

A Simple Network

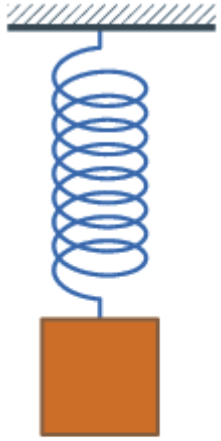
Input: Mass or M (kg)
Output: Length or L (m)



M	L
Input	Output
0.125	0.39
0.25	0.40
0.5	0.43
1	0.48
2	0.58
3	???

Based on Mary Attenborough, in Mathematics for Electrical Engineering and Computing, 2003

A Simple Network



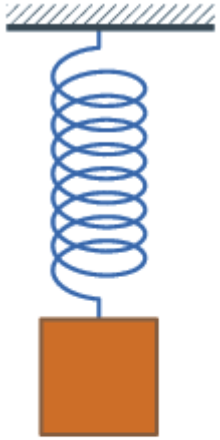
M	L
0.125	0.39
0.25	0.40
0.5	0.43
1	0.48
2	0.58
3	0.68

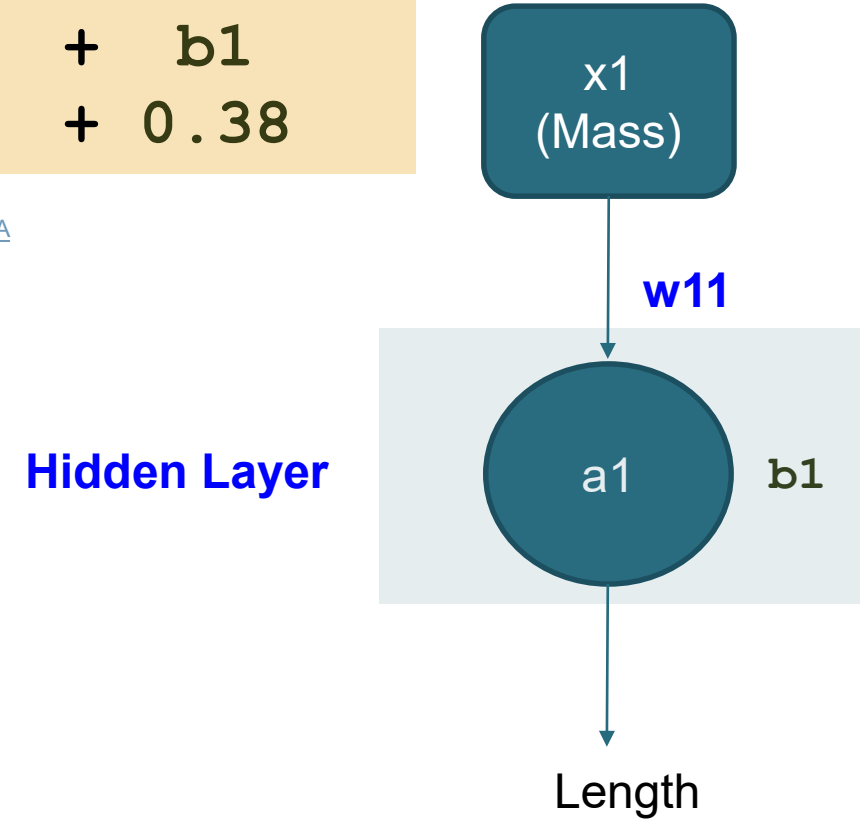
```
L = 0.1 * Mass + 0.38
```


Mary Attenborough, in Mathematics for Electrical Engineering and Computing, 2003

A Simple Network


```
a1 = x1   * w11   +  b1
 L = M    * 0.1   + 0.38
```



Hidden Layer

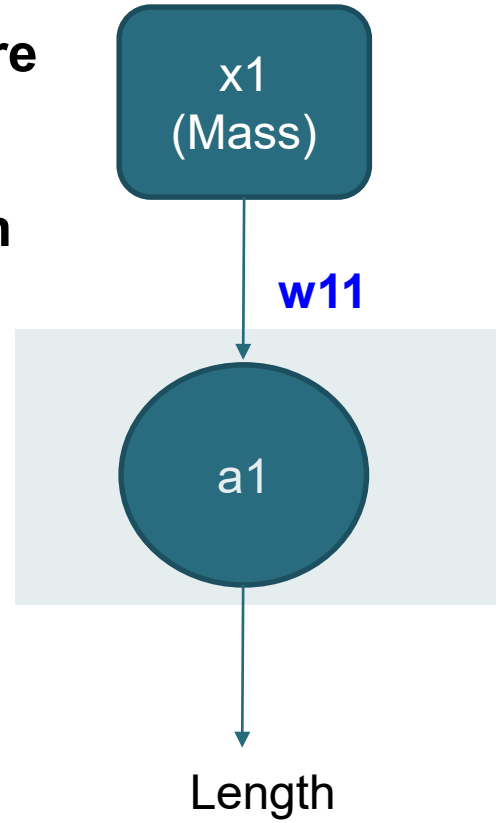
M	L
0.125	0.39
0.25	0.40
0.5	0.43
1	0.48
2	0.58
3	0.68

```
These are the model variables: [array([[0.10058284]], dtype=float32), array([0.37793916], dtype=float32)]
```

Based on Mary Attenborough, in Mathematics for Electrical Engineering and Computing, 2003

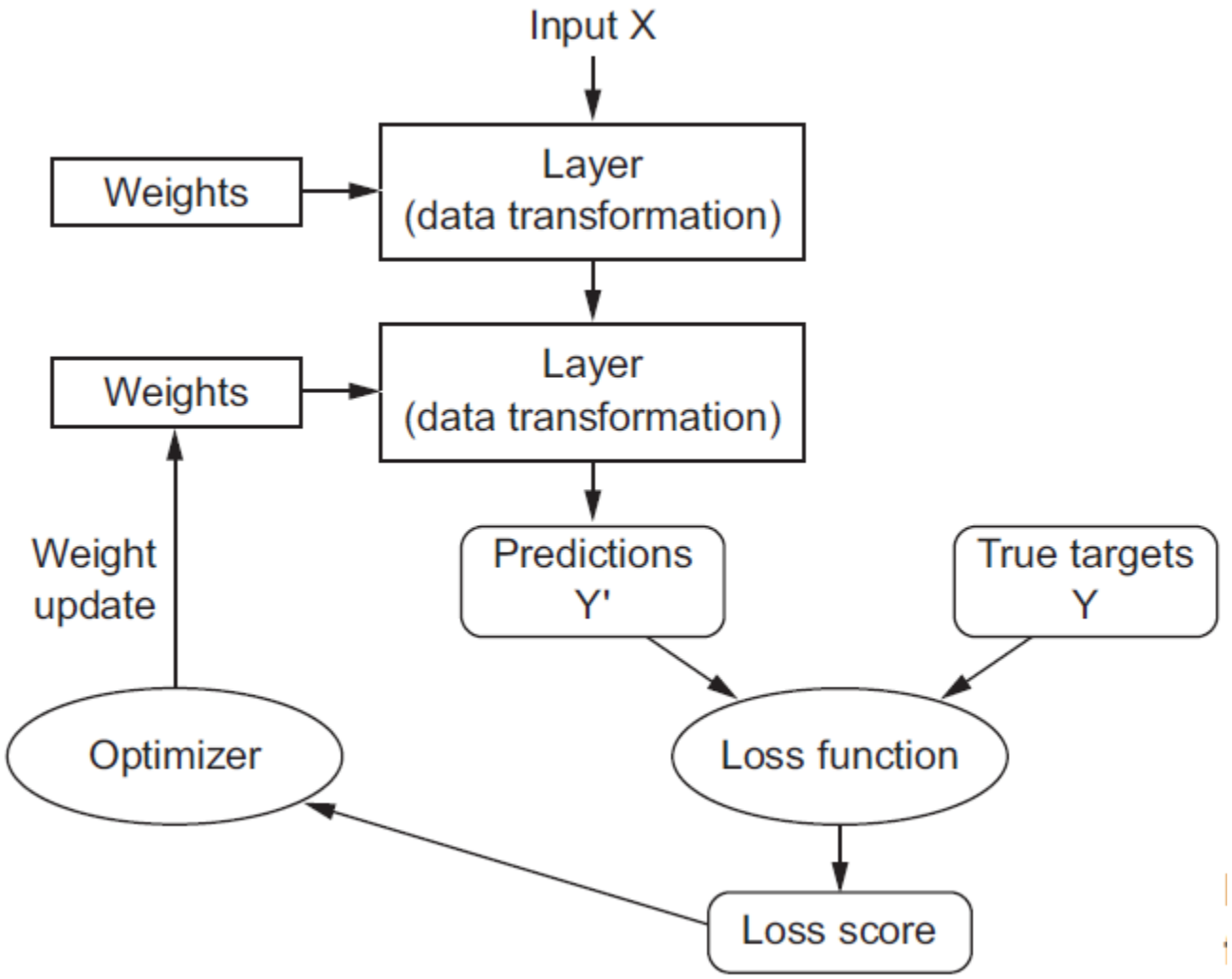
Error minimization

- Goal is to choose Ws such that predictions of the network should be close to y
- Error function or cost function a measure how good our predictions are
- Eventually, we want to pick a set of w that minimizes the error function



Deep Learning Procedure

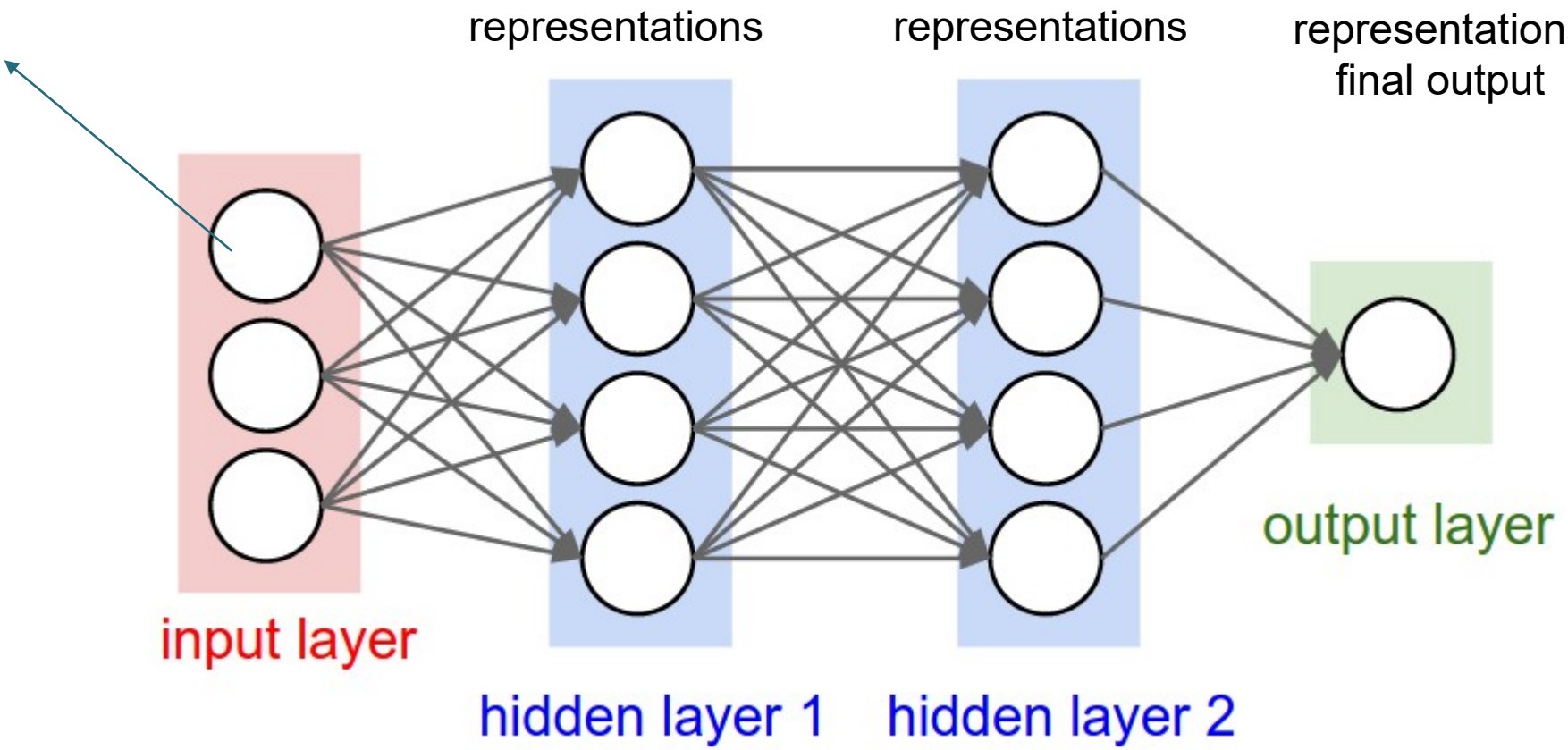
Taken from Deep Learning with Keras book



Vanilla network

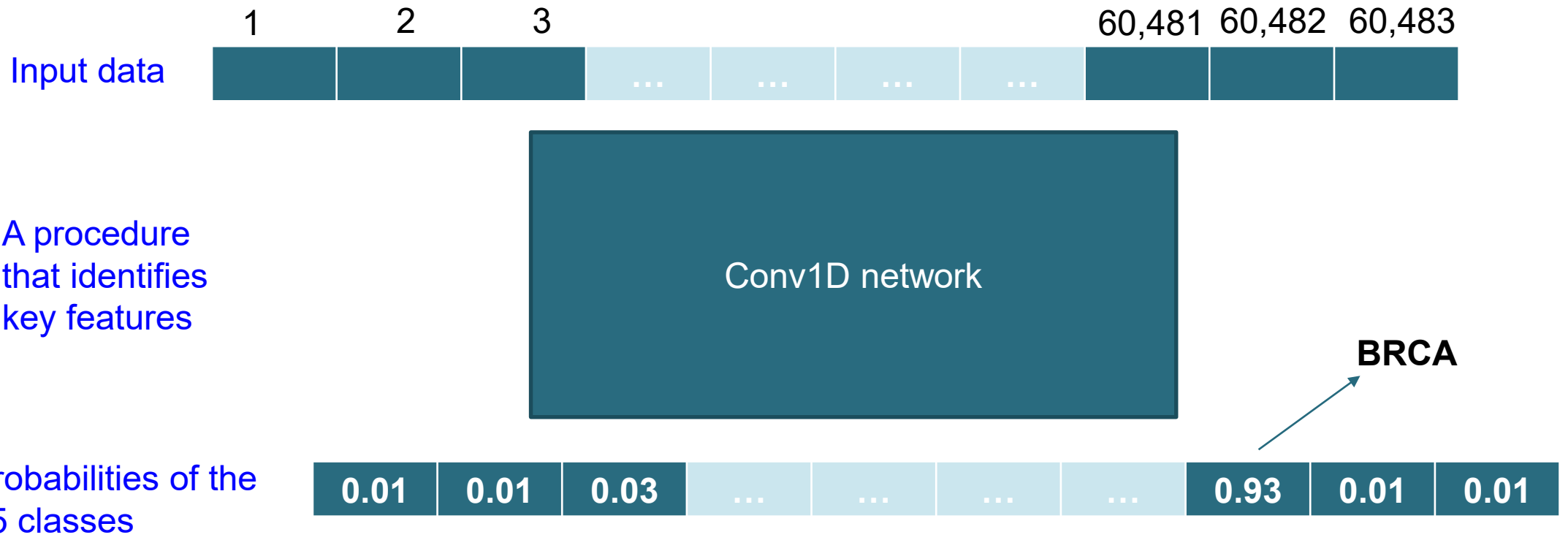
Each neuron receives input from all the neurons in the previous layer (densely connected)

Neuron: a unit that holds a number



Convolutional Neural Network

- **We are going to take a vector of genomic expression values and feed them into a network with a series of operations to create a model**
- **Model is what we call convolutional-1D network**

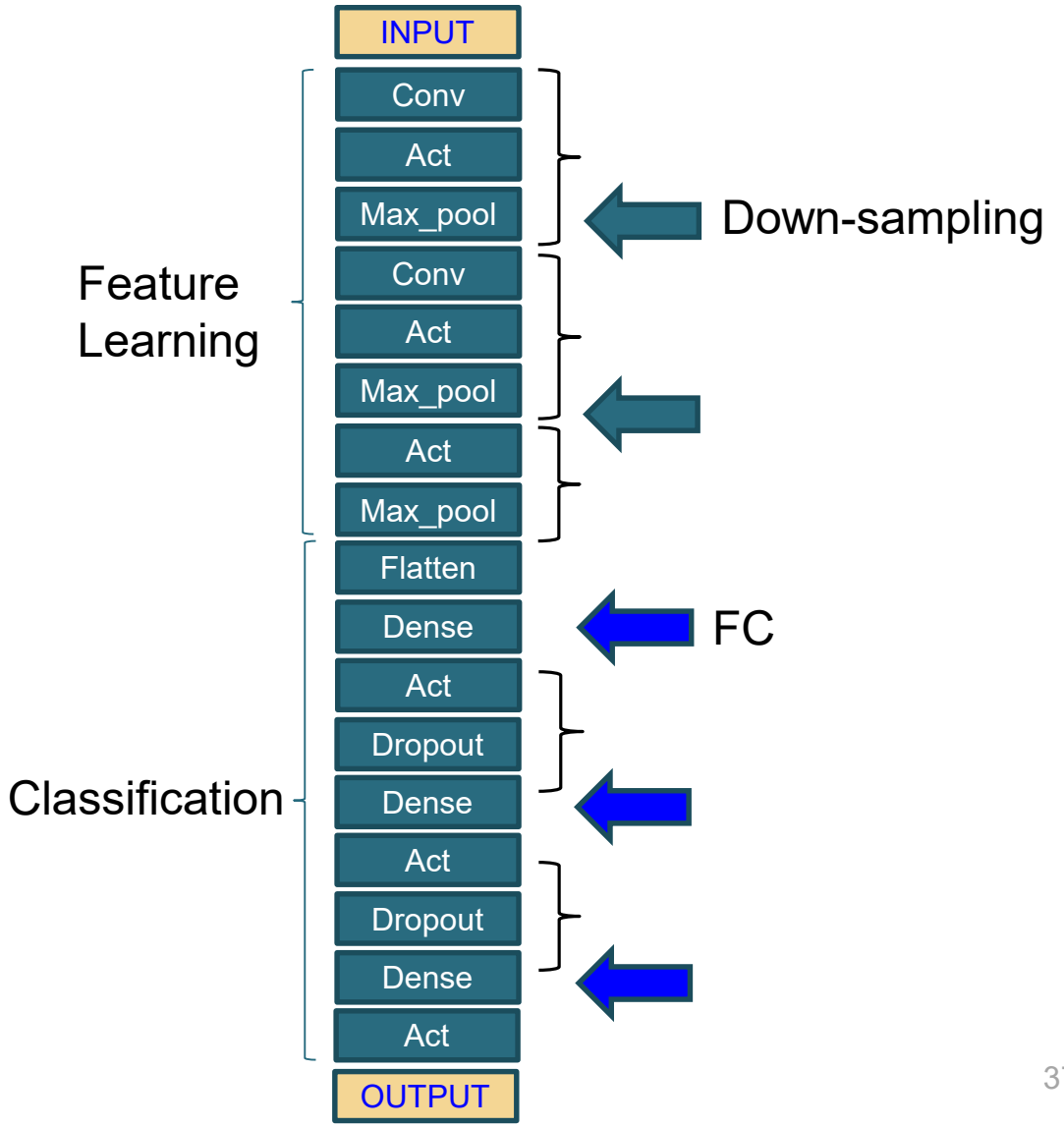


Components of conv1D

1. **Act: Activation**
2. **Conv: Convolution**
3. **Max_pool: Maxpooling**
4. **Flatten**
5. **Dense**
6. **Dropout**

Topology of a network defines a “hypothesis space”

Choosing a specific topology is usually not straight-forward and comes with practice (& domain knowledge).



ConvNets Architecture

- **Depends on the problem**
- **Try that worked for a similar problem before you try new options**
- **[(CONV-RELU) * N - POOL?] * M - (FC-RELU)*K, SOFTMAX**
  - N is usually up to ~5
  - M is large
  - 0 <= K <= 2.
- **Trend is to use smaller filter and deeper architectures**
    - *Fei-Fei Li & Justin Johnson & Serena Yeung Lecture notes*

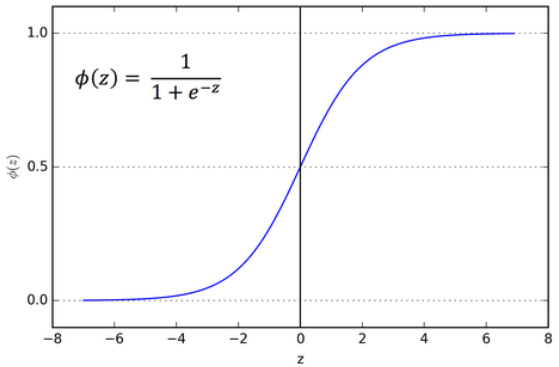
1. Activation Function

1. **Act: Activation**
2. Conv: Convolution
3. Max_pool: Maxpooling
4. Flatten
5. Dense
6. Dropout

- **Activation functions are included to create non-linearity**

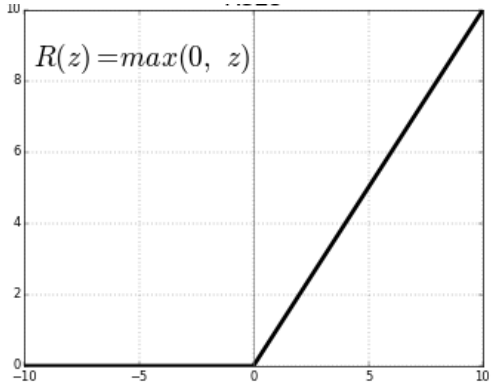
- **Sigmoid**
- **ReLU**
- **Leaky ReLU**
- **ELU**
- **Maxout**
- **Tanh**

Sigmoid

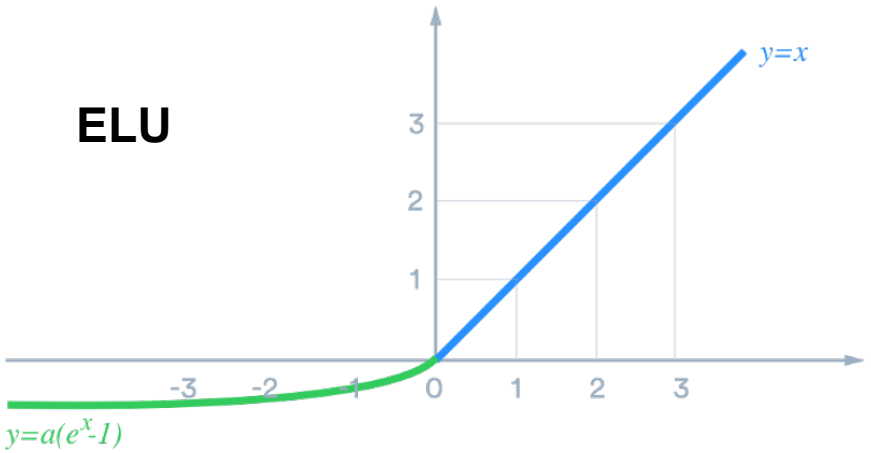


Squashes the #s to [0, 1]

ReLU

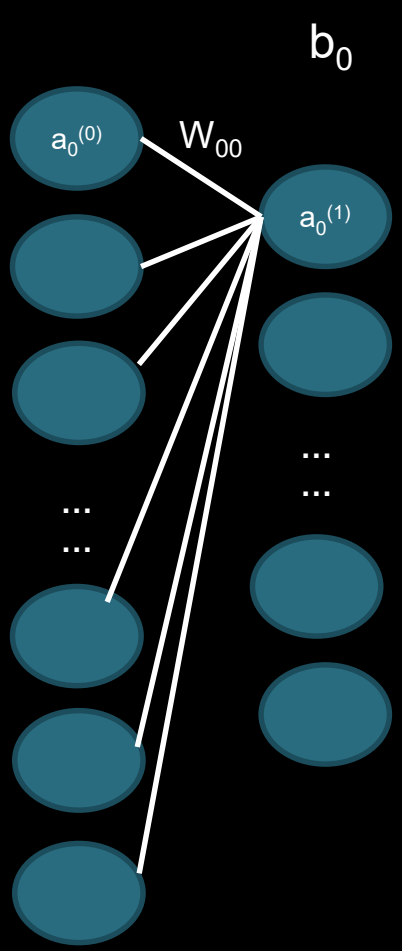


ELU



1. Activation function

$$a^{(L)} = \text{ReLU}(w^{(L)}a^{(L-1)} - b^{(L)})$$



$$\text{ReLU}\left(\begin{bmatrix} W_{0,0} & W_{0,1} & \cdots & W_{0,n} \\ W_{1,0} & W_{1,1} & \cdots & W_{1,n} \\ \vdots & \vdots & \vdots & \vdots \\ W_{k,0} & W_{k,1} & \cdots & W_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix} \right)$$

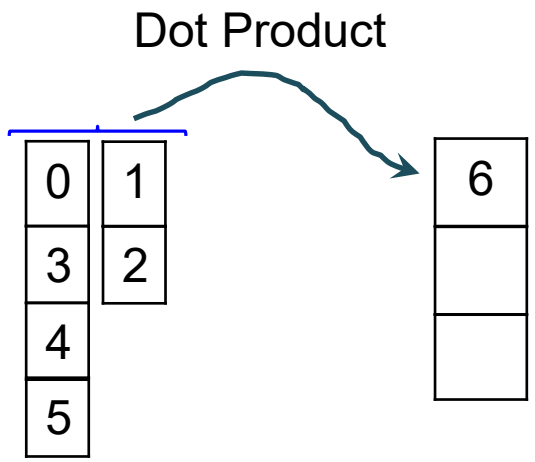
$$a_0^{(1)} = \text{ReLU}(W_{00}a_0^{(0)} + W_{0,1}a_1^{(0)} + \ldots W_{0,n}a_n^{(0)} - b_0))$$

2. Convolution

1. **Act: Activation**
2. Conv: Convolution
3. Max_pool: Maxpooling
4. Flatten
5. Dense
6. Dropout

Process of applying filter (kernel) to the data for the purpose of subsampling. Kernel is a matrix that has a smaller dimension than the input data creates chunks

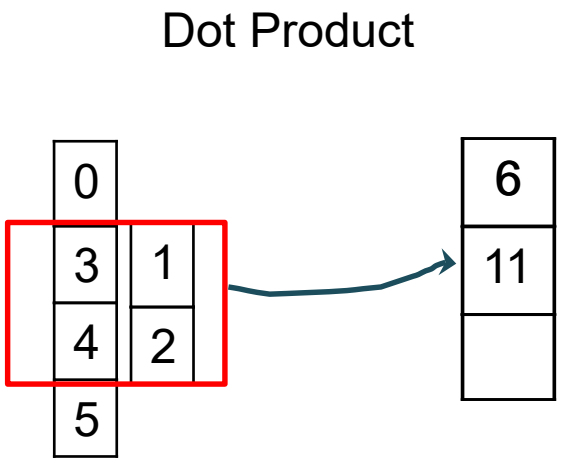
Reduces the number of parameters and allow creation of deeper networks



2. Convolution

Process of applying filter (kernel) to the data for the purpose of subsampling. Kernel is a matrix that has a smaller dimension than the input data creates chunks

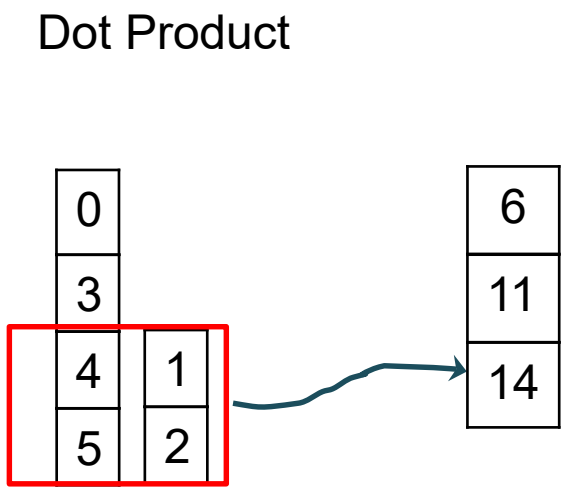
Reduces the number of parameters and allow creation of deeper networks



2. Convolution

Process of applying filter (kernel) to the data for the purpose of subsampling. Kernel is a matrix that has a smaller dimension than the input data creates chunks

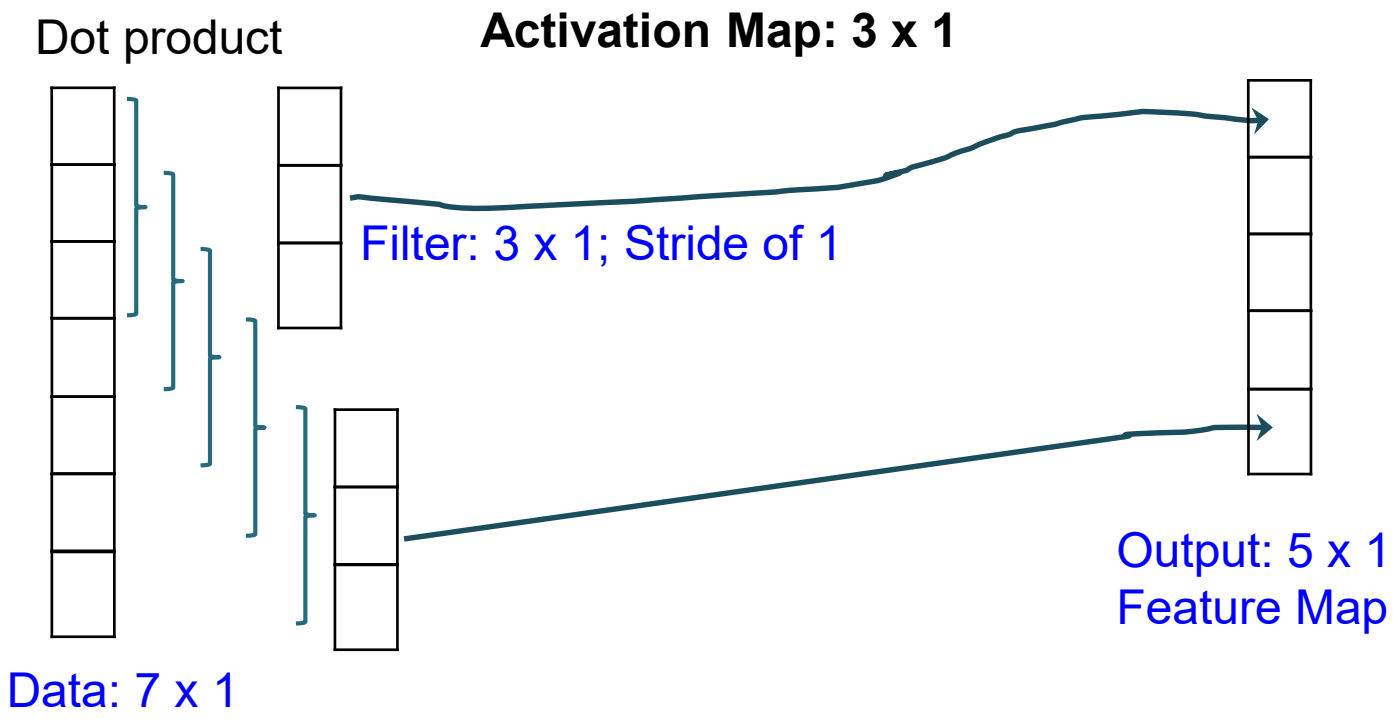
Reduces the number of parameters and allow creation of deeper networks



2. Convolution

Process of applying filter (kernel) to the data for the purpose of subsampling. Kernel is a matrix that has a smaller dimension than the input data creates chunks

Reduces the number of parameters and allow creation of deeper networks



((N-F)/stride) + 1 will be the size after filtering

(7-3)/1+1 = 5 ;
zero padding on the border

2. Convolution

- **Summary**
- **Common settings**
  - Number of filters (K): Chosen in powers of 2 (ex. 32, 64, etc.)
  - Spatial Extent (F): 3 or 5
  - Stride (S): 1 or 2
  - Zero padding (P): 0, 1, 2

2. Convolution

- **Convolution Layer**
  - Hyperparameters
    - Number of filters
    - Spatial extent
    - Stride
    - Amount of zero padding

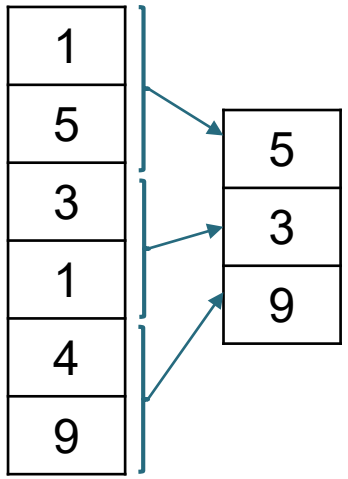
Andrew, an expert in CANDLE, can help you with Hyperparameter optimization.

andrew.weisman@nih.gov

3. Pooling

1. **Act: Activation**
2. Conv: Convolution
3. Max_pool: Maxpooling
4. Flatten
5. Dense
6. Dropout

- **Pooling makes the representations smaller/manageable (downsampling) by retaining only important features; creates smaller clusters of manageable size**
- **Each activation map will be pooled separately.**
- **Common approach is Max Pooling**



Max-pooling with filter size of 2x1 and stride of 2

Max Pooling Intuition:
Enhancing the signals by looking at a region and pick the maximum activation value

Each of these are activation that we are looking for

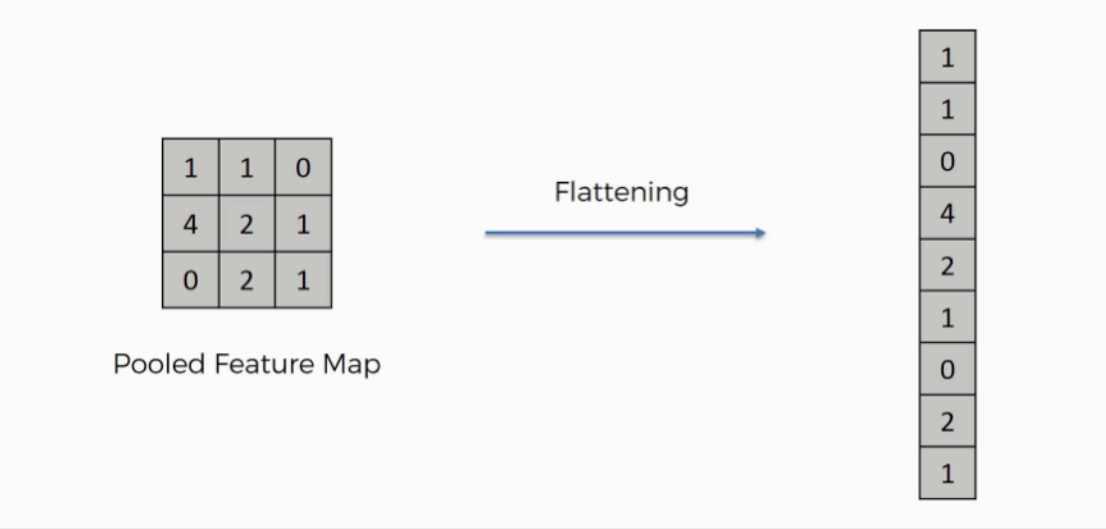
Research shows that zero-padding is not followed. Because we are interested in down-sampling

Common setting for filter 2 or 3

4. Flatten

1. **Act: Activation**
2. Conv: Convolution
3. Max_pool: Maxpooling
4. Flatten
5. Dense
6. Dropout

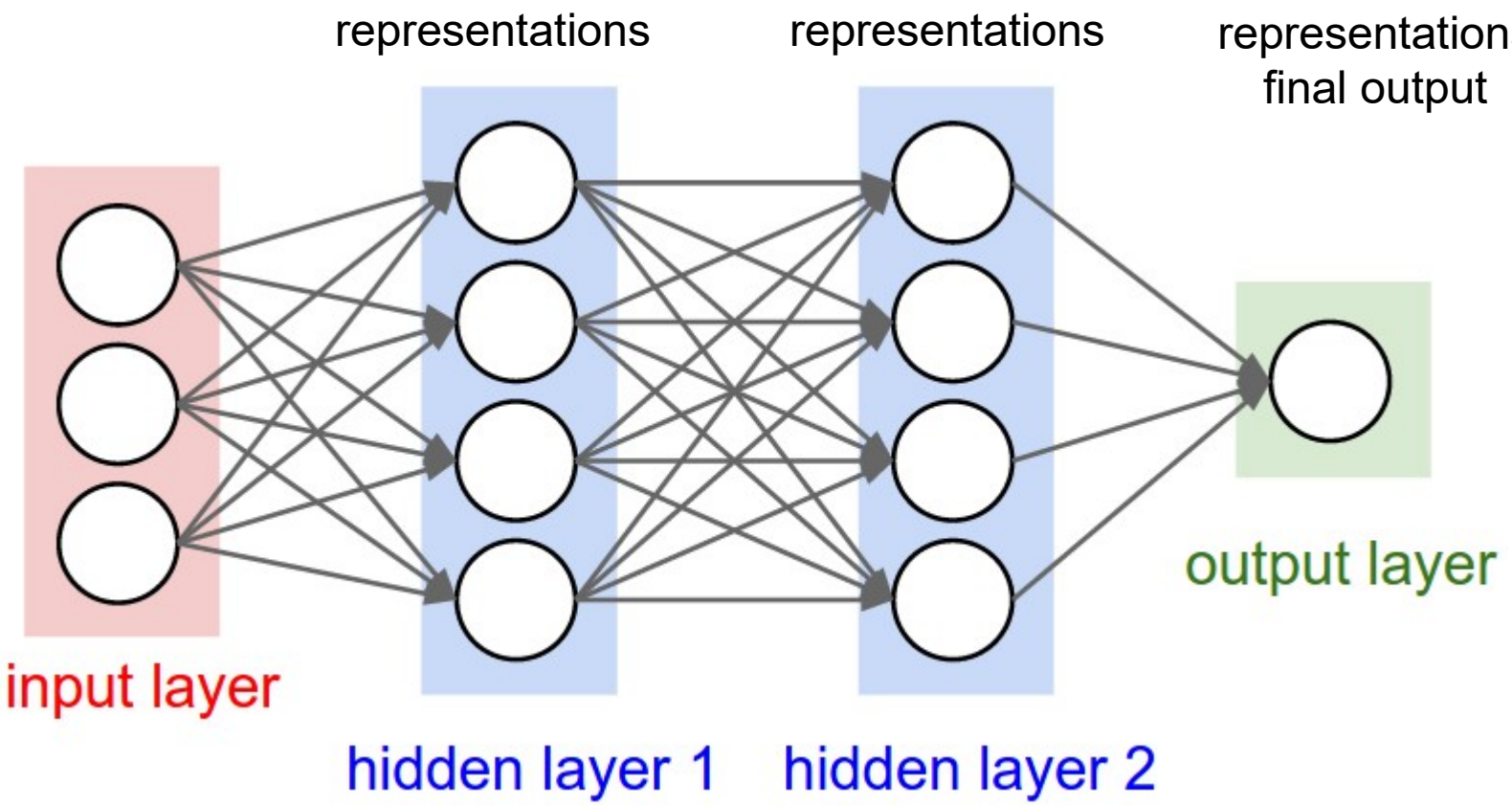
Procedure to transform a 2D matrix (features) to a 1D vector which in turn can be fed into a fully-connected layer (dense)



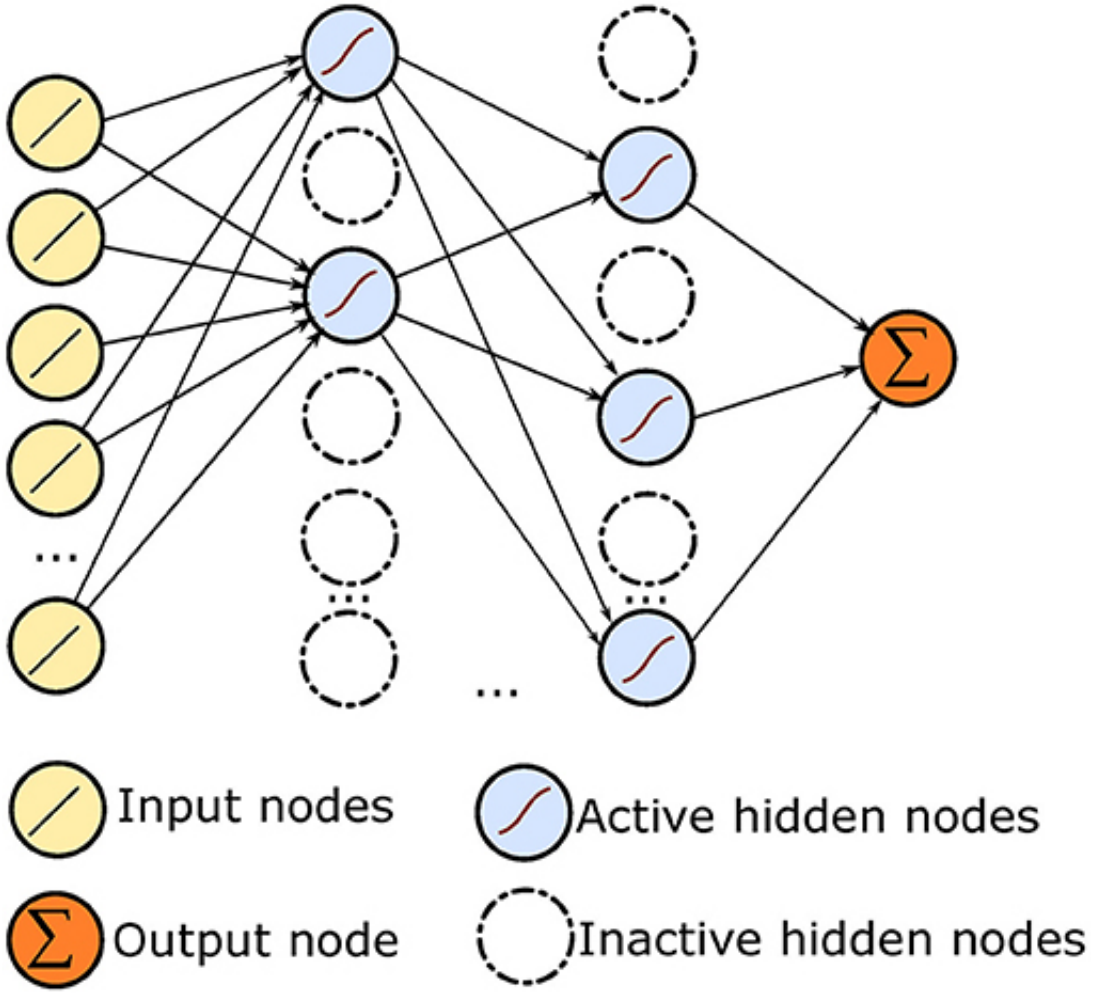
5. Dense

1. **Act: Activation**
2. Conv: Convolution
3. Max_pool: Maxpooling
4. Flatten
5. Dense
6. Dropout

Each neuron receives input from all the neurons in the previous layer (densely connected)



6. Dropout



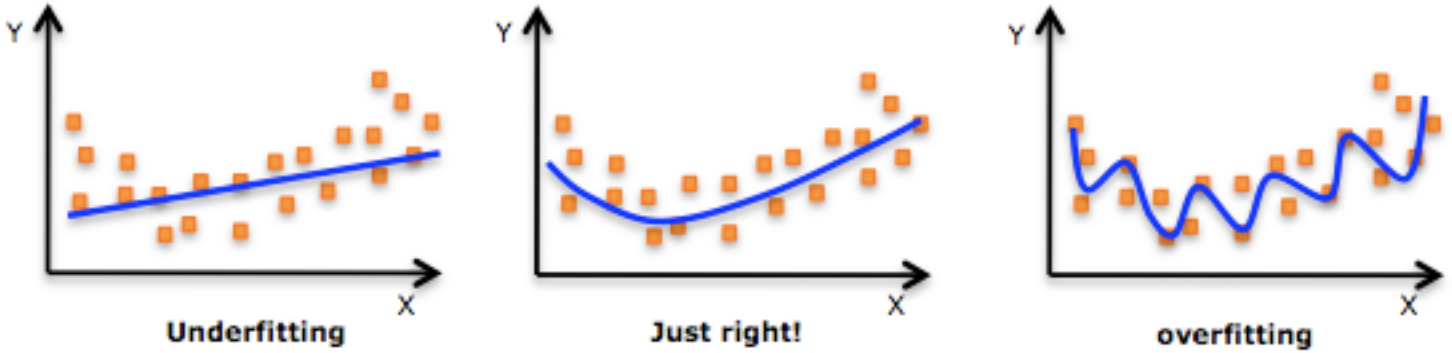
Imbalance in the weights among the nodes can lead to some node weights not contributing to the learning

One solution:
Remove a random proportion of selection of neurons in a neural network during training

Can help weak learners become strong learners

6. Dropout

1. **Act: Activation**
2. Conv: Convolution
3. Max_pool: Maxpooling
4. Flatten
5. Dense
6. Dropout

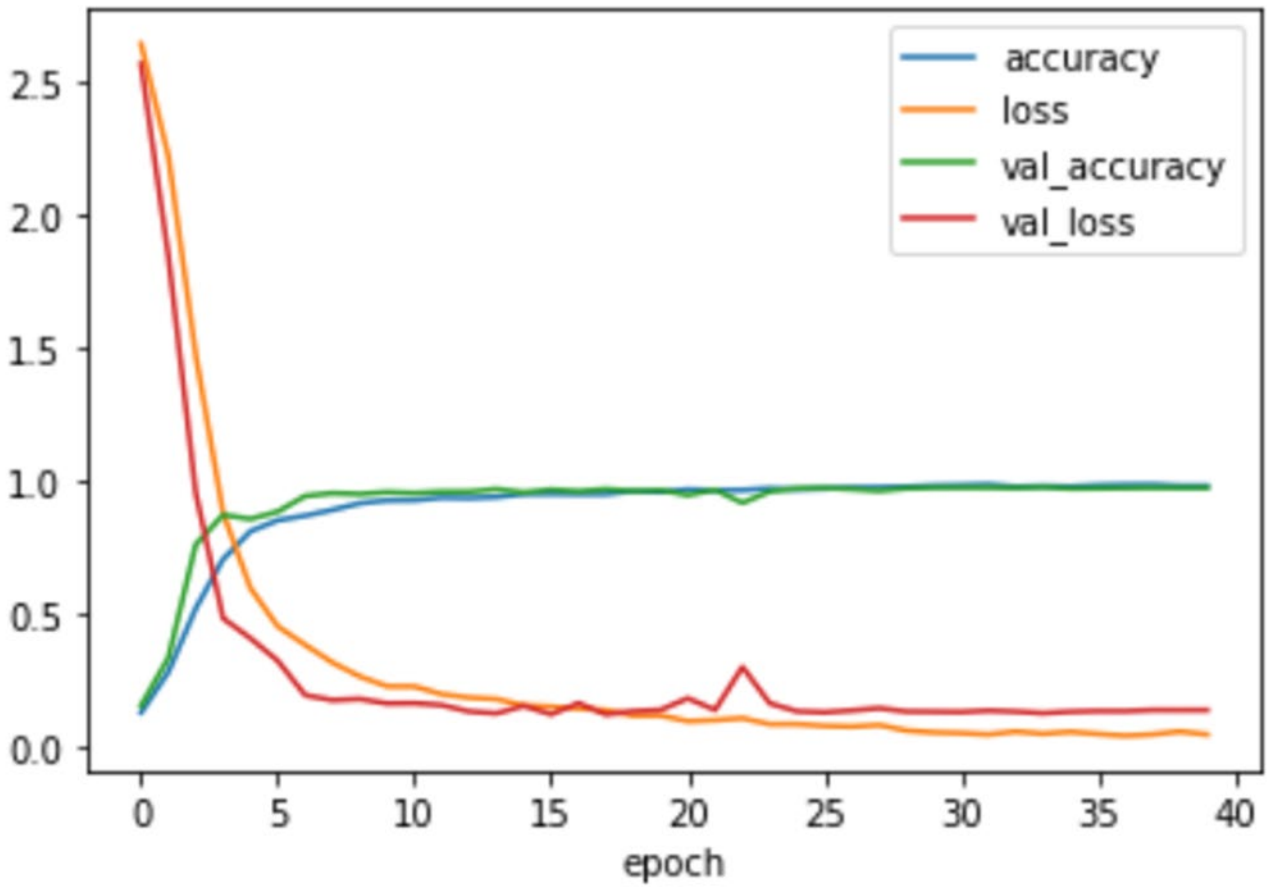


Model Summary


```
1.0 128 10 1
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 60464, 128)        2688
_________________________________________________________________
activation_1 (Activation)    (None, 60464, 128)        0
_________________________________________________________________
max_pooling1d_1 (MaxPooling1 (None, 60464, 128)        0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 60455, 128)        163968
_________________________________________________________________
activation_2 (Activation)    (None, 60455, 128)        0
_________________________________________________________________
max_pooling1d_2 (MaxPooling1 (None, 6045, 128)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 773760)            0
_________________________________________________________________
dense_1 (Dense)              (None, 200)               154752200
_________________________________________________________________
activation_3 (Activation)    (None, 200)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 200)               0
_________________________________________________________________
dense_2 (Dense)              (None, 20)                4020
_________________________________________________________________
activation_4 (Activation)    (None, 20)                0
_________________________________________________________________
dropout_2 (Dropout)          (None, 20)                0
_________________________________________________________________
dense_3 (Dense)              (None, 15)                315
_________________________________________________________________
activation_5 (Activation)    (None, 15)                0
=================================================================
Total params: 154,923,191
Trainable params: 154,923,191
Non-trainable params: 0
```

~ 154 M parameters

Model Performance



Inference

- **Key points about dataset to note**
  - Same dimension (feature) as the input data
    - **Keras:** Make sure the shape is the same as the training data
  - Same scaling as the training data

After the Workshop…

Please answer a few, short questions.

Check your email in-box!

Thank you!

Thank you!

Questions/Comments

S. Ravichandran
ravichandrans@mail.nih.gov