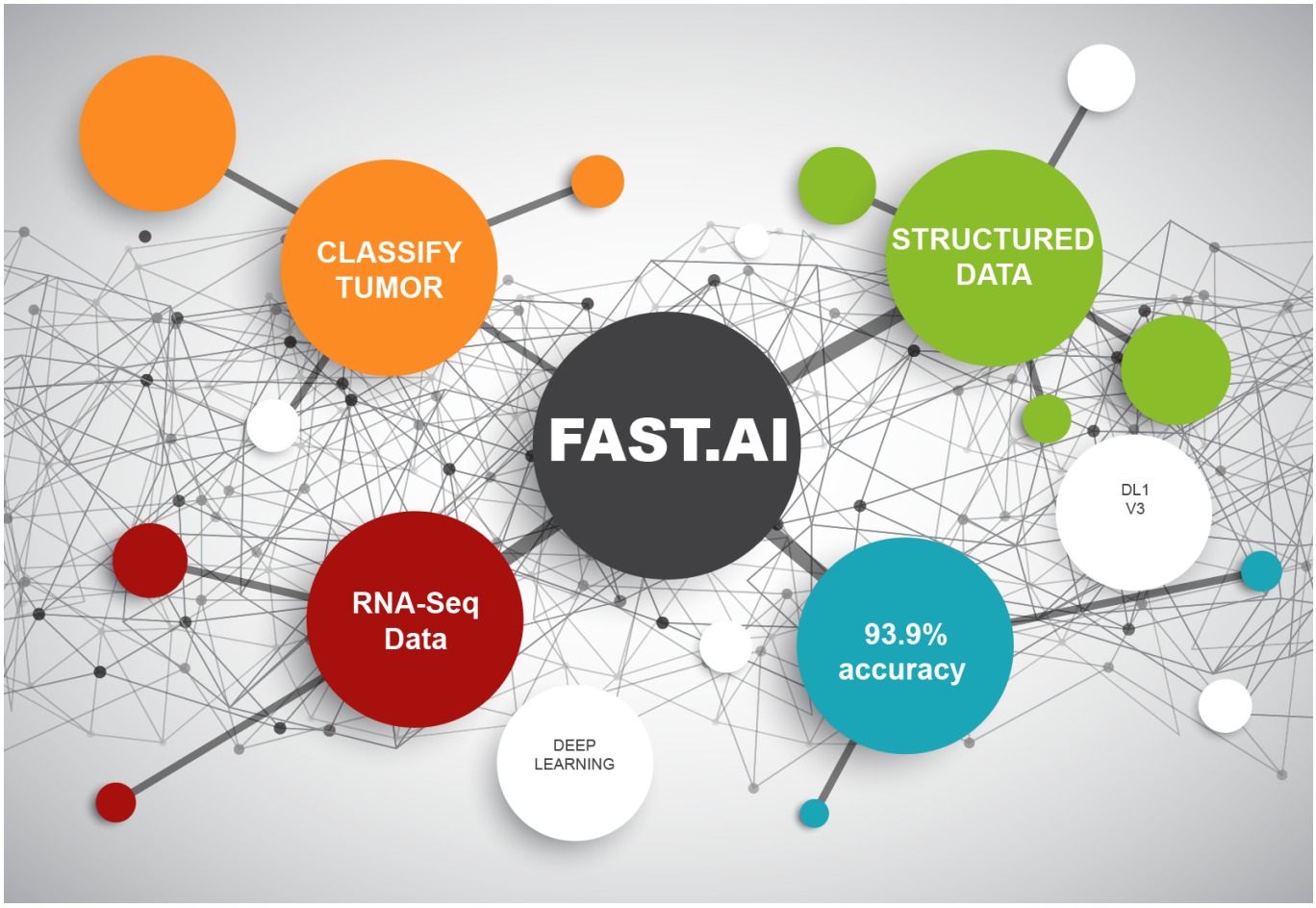
Tumor Classification using Gene Expression Data — poking at a problem using Fast.AI again

Alena Harley Follow
Nov 12, 2018 · 3 min read



Recently high-throughput RNA sequencing (RNA-seq) has become the dominant method for studying gene expression. Large amount of gene expression data have been generated in the field of cancer genomics, *e.g.* The Cancer Genome Atlas (TCGA), where gene expression data for 9,784 tumors is available. ARCHS4 — a web resource that provides expression data at the gene and transcript levels — contains recently recomputed TCGA RNA-Seq data that has been processed using the same pipeline to

remove batch effects that inadvertently originate from sequencing samples at different laboratories.

In 2017, a paper on classifying tumor samples based on RNA-Seq data has made headlines, see RNA-Seq Blog. The authors classified TCGA RNA-Seq samples into **31 classes** with **overall accuracy of ~90%** using genetic algorithm as the gene/feature selection method and the *k*-nearest neighbors algorithm.

Here, I have tried to replicate the results of this paper using Fast.AI library for **33 tumor classes** with overall **accuracy of 93.9%**.

In addition to log2(TPM+0.001) expression values computed per gene, I added result of pathway enrichment analysis for 50 Cancer Hallmark pathways as categorical variables.

I used Fast.AI Categorical Embedding, where the creation of embeddings for categorical variables is performed while training the network end-to-end on structured (tabular) data. The embedding captures the relationships between categories better than popular one-hot-encoding.

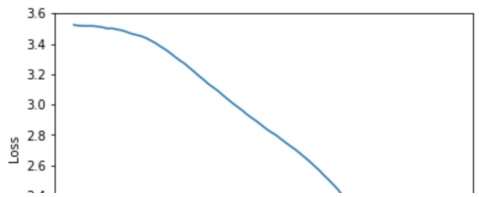
A simple feed-forward neural network model with two hidden layers is constructed using Fast.AI library. While training, I found that the network needed further regularization and tuned the dropout levels for each layer.

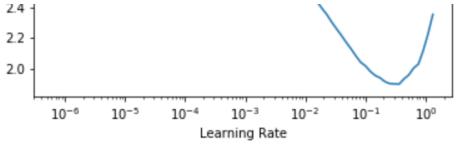
```
In [143]: learn = get_tabular_learner(data,layers=[500,50], metrics=accuracy,ps=[.2,.1])
```

data is an instance of **TabularDataBunch** that contains stratified data for TCGA samples (70%/30% train/test split)

Learning rate is chosen using Leslie N. Smith's method implemented in Fast.AI repo — right before the loss starts increasing and, preferably, at the point of its greatest decline (0.05).

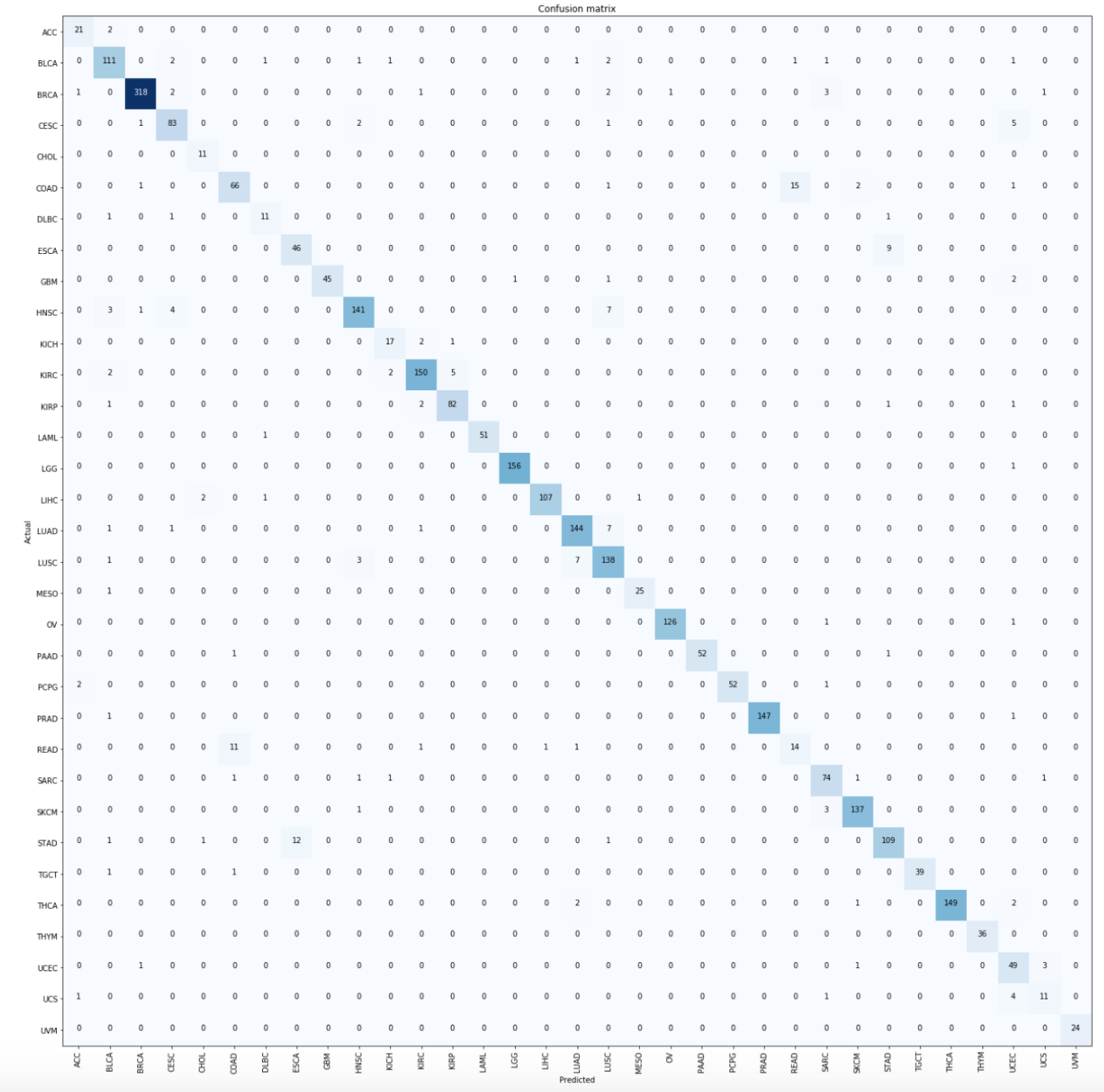
```
In [145]: lrf=learn.lr_find()
          learn.recorder.plot()

          LR Finder complete, type {learner_name}.recorder.plot() to see the graph.
```





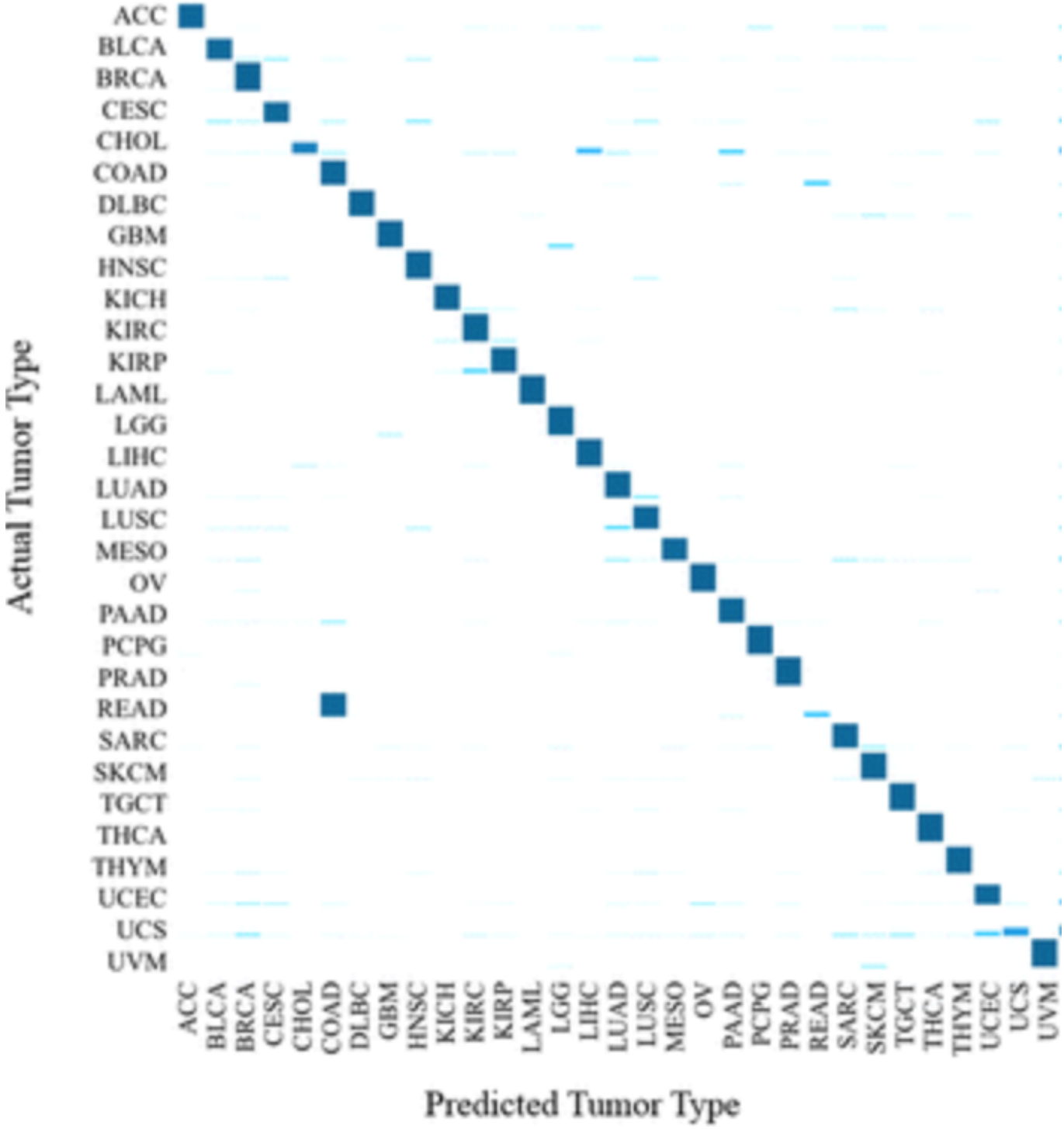
learning rate finder

Let's visualize the performance of our algorithm using confusion matrix:



Confusion matrix for 33 TCGA tumor classes. **Accuracy achieved is 93.9%.**

Confusion matrix from the original paper looks quite similar.



Our mis-classifications as well as misclassifications of the original paper are primarily within the same organ systems, *e.g.* colon(COAD) and rectal (READ) cancer; stomach (STAD) and esophageal (ESCA) cancer; cervical (CESC) and endometrial (UCEC) cancer.

Fast.AI is an amazing resource, I believe any researcher or a deep learning practitioner should take a closer look at Fast.AI.

Thank you Jeremy Howard and Rachel Thomas for creating this amazing resource!

Machine Learning