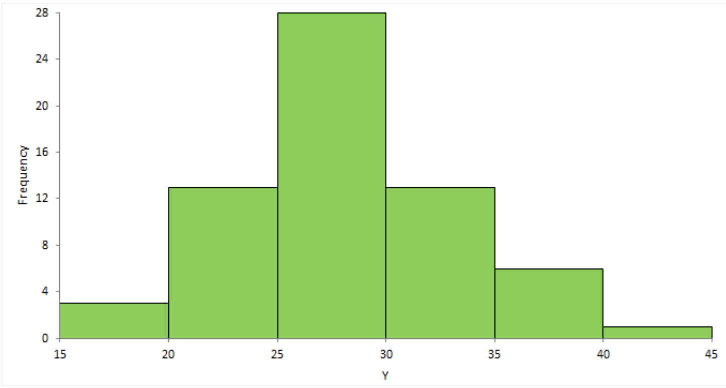
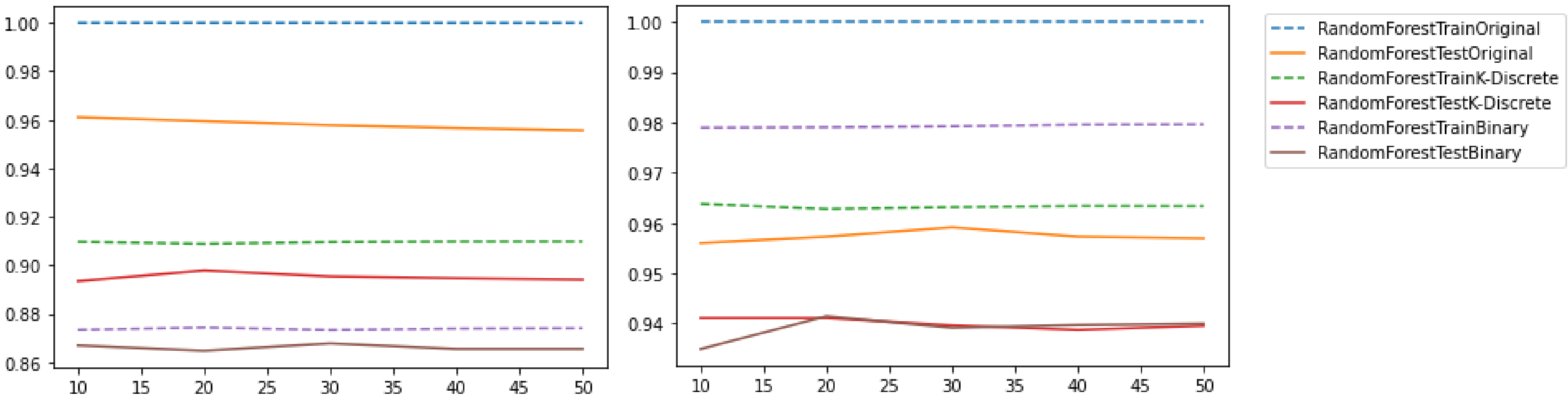
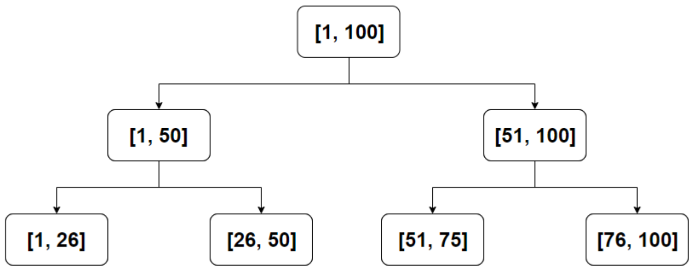
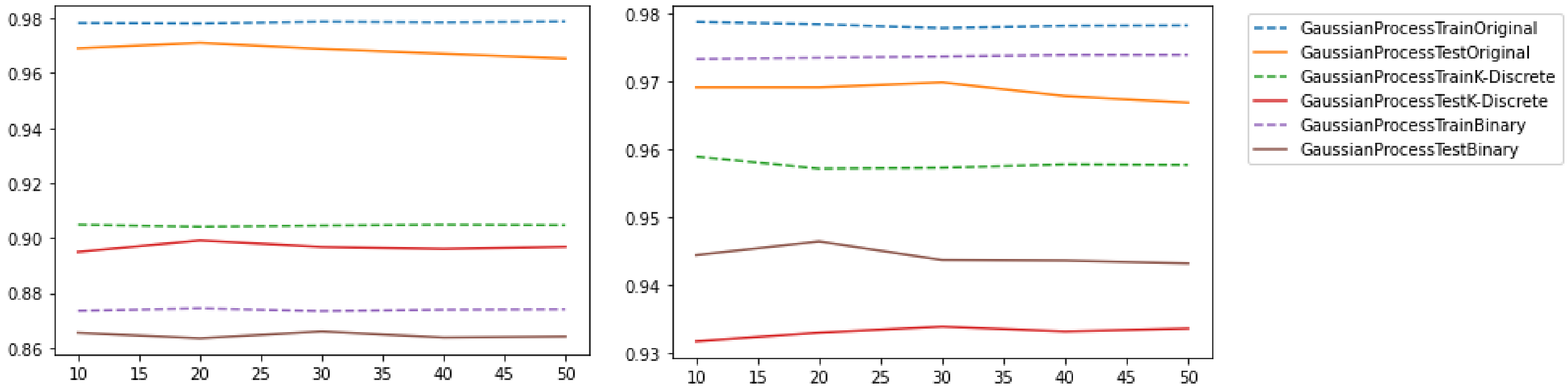






Ex: “27” = 01, “67”=10, “88”=11

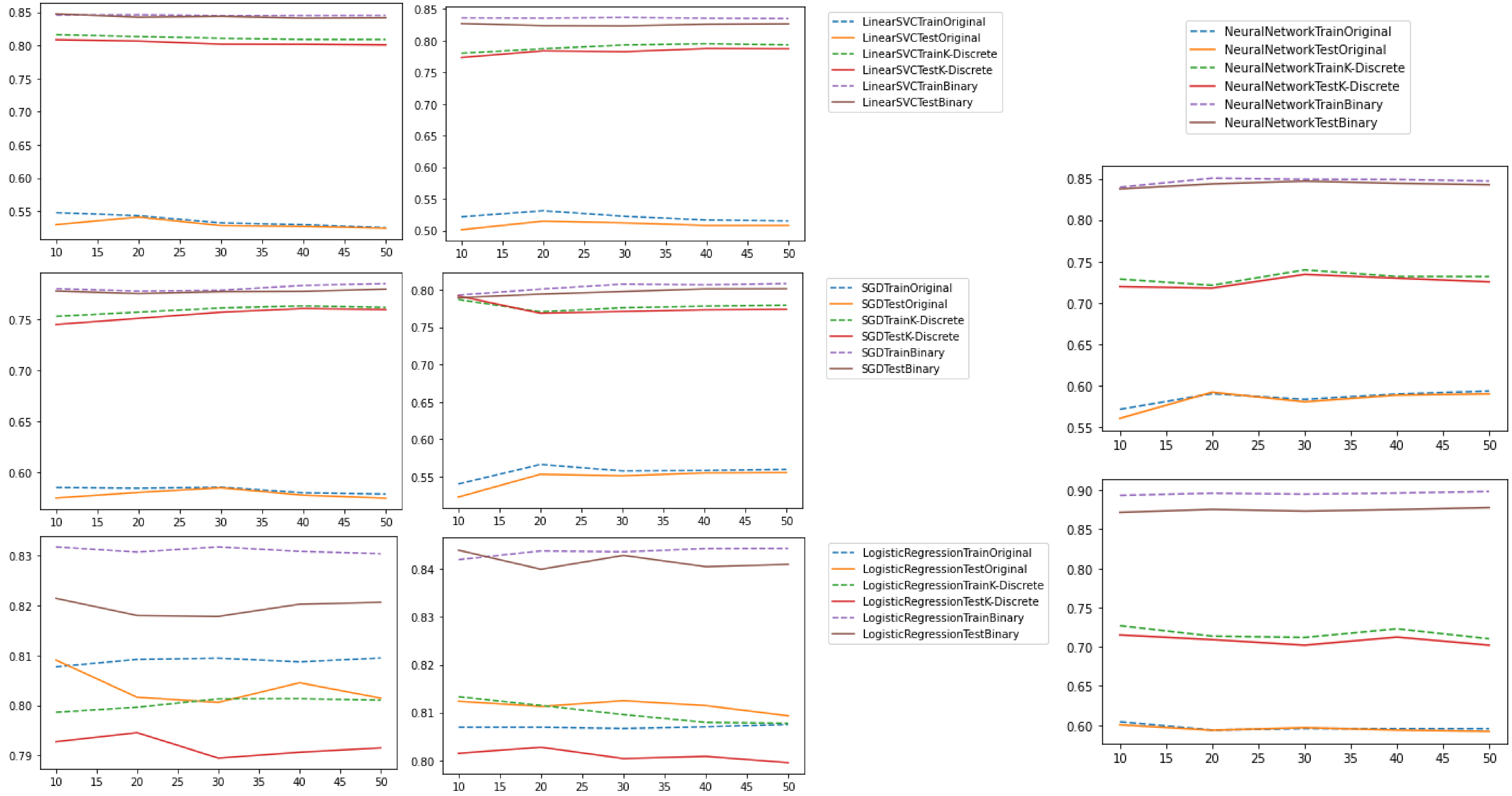


Feature Transformation in ML Discretization & Binarization

YIJIE (EJAY) GUO

YU CHEN

EJAY: YG2418@NYU.EDU YU:YC4902@NYU.EDU



EJAY: YG2418@NYU.EDU YU:YC4902@NYU.EDU

Noise Reduction

[1] Linear Models are more sensitive towards out layers, noises and fluctuates in the data. The discretization and binarization improve signal-to-noise ratio and smooth out the noises.

[2] Non-linear models are not so sensitive towards the noises and very likely to be more overfitting.

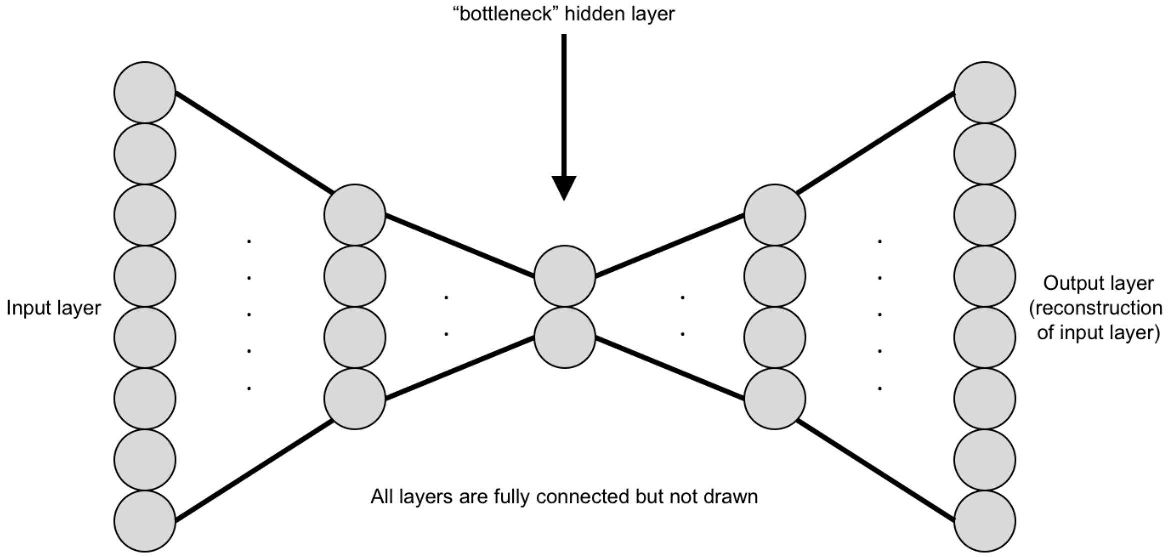
Quantization

[1] More Model Choices

[2] Memory and Computation Reduction

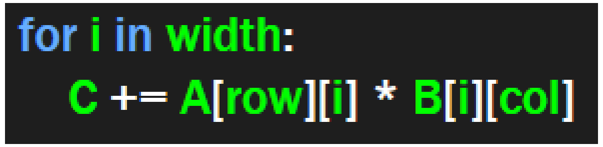
[3] Compressing the floating point input/output values in DNN to a fixed point representation, like 8-bit or 16-bit integers.

Information Bottleneck

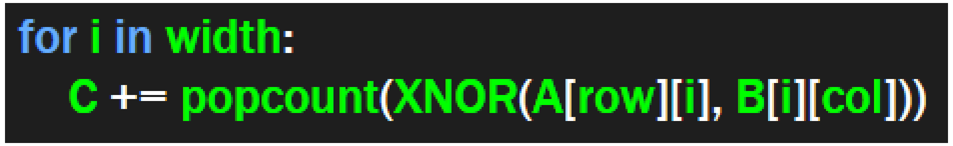

"bottleneck" hidden layer
Input layer
Output layer (reconstruction of input layer)
All layers are fully connected but not drawn

Binary Neural Network

Inputs, outputs and weights are all binary values. By binary here, we mean Bipolar Binary, i.e. +1 & -1 values.



```
for i in width:
   C += A[row][i] * B[i][col]
```



```
for i in width:
   C += popcount(XNOR(A[row][i], B[i][col]))
```