Eksplorasi Hyperparameter CNN dan Neural Network

Imam Ekowicaksono - (NIM. 33220306)

1 Pendahuluan

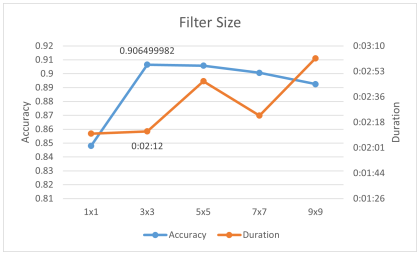
Persoalan klasifikasi merupakan salah satu persoalan populer yang ada pada topik pembelajaran mesin (*machine learning*). Persoalan klasifikasi ini dapat ditangani dengan menggunakan metode klasifikasi yang berbasis neural network dengan mengklasifikasi data gambar Fashion MNIST.

2 Klasifikasi

Persoalan klasifikasi ini menggunakan dataset Fashion MNIST untuk dilakukan training dan testing. Fashion-MNIST adalah kumpulan data gambar traning sebanyak 60.000 sampel dan testing sebanyak 10.000 sampel. Setiap sampel terdiri dari gambar grayscale berukuran 28x28 dan memiliki 10 kelas label.

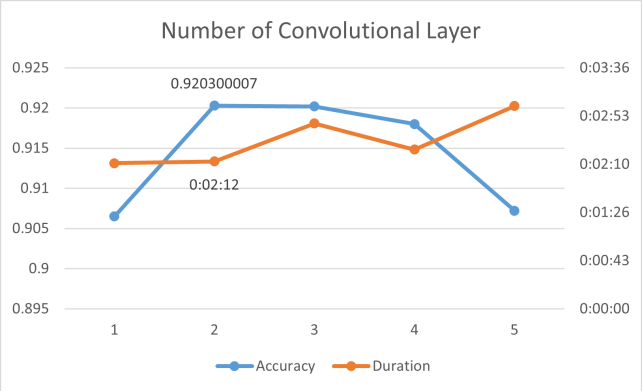
Pada persoalan klasifikasi ini, dilakukan perancangan model CNN dengan menggunakan keras. Keras adalah library untuk membuat model CNN. Keras memiliki beberapa library yang bisa digunakan untuk membuat model CNN. Model terbaik yang dapat diselesaikan sampai dengan saat ini menggunakan 2 convolutional layer dan 2 fully connected layer.

Percobaan dilakukan dengan mengembangkan prebuild model handwritten digit recognition model yang sudah ada. Percobaan pertama dilakukan dengan mengecek ukuran kernel yang optimal untuk menghasilkan model CNN yang optimal. Hasil yang optimal diperoleh dengan menggunakan kernel berukuran 3×3. Hasil eksplorasi disajikan pada gambar 1 berikut:



Gambar 1: Akurasi berdasarkan ukuran kernel

Setelah mendapatkan ukuran kernel yang optimal, langkah berikutnya adalah mengecek jumlah convolutional layer yang optimal. Percobaan pertama dilakukan dengan menggunakan 2 convolutional layer. Percobaan berikutnya dilakukan dengan 3 convolutional layer sampai dengan 5 convolutional layer. Hasil eksplorasi disajikan pada gambar 2 berikut:



Gambar 2: Akurasi berdasarkan jumlah convolutional layer

Pada percobaan kedua ini diperoler jumlah convolutional layer yang optimal adalah 2 convolutional layer dengan akurasi sebesar 90.65%.

Percobaan berikutnya adalah mengecek jumlah kernel pada setiap convolutional layer yang optimal. Percobaan dilakukan dengan mengkombinasi seluruh kemungkinan jumlah kernel pada setiap convolutional layer. Hasil eksplorasi disajikan pada gambar 3 berikut:

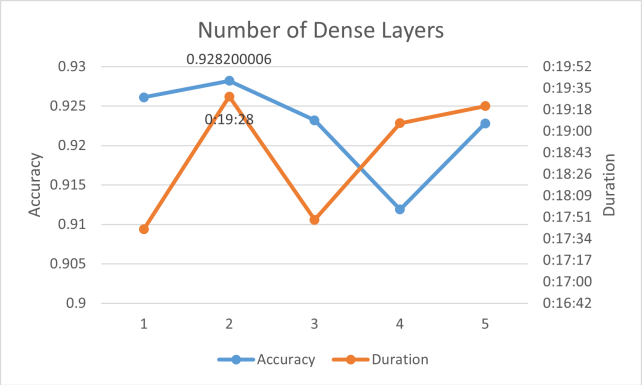
Accuracy						
		Conv-2				
		24	32	64	128	256
Conv-1	24	92.03%	91.73%	92.49%	92.13%	92.48%
	32	91.91%	91.17%	92.23%	**92.61%**	92.44%
	64	91.56%	92.28%	92.23%	91.78%	92.51%
	128	92.27%	92.13%	92.41%	92.43%	92.01%
	256	91.78%	92.01%	92.39%	91.88%	92.20%

Gambar 3: Akurasi dari setiap kemungkinan jumlah kernel pada setiap convolutional layer

Percobaan ini menghasilkan akurasi terbaik pada 92.61% dengan jumlah kernel sebesar 32 pada convolutional layer pertama dan 128 kernel pada convolutional layer kedua.

Percebaan berikutnya adalah mengecek jumlah dense layer yang optimal. Percobaan pertama dilakukan dengan menggunakan 2 dense layer. Percobaan

berikutnya dilakukan dengan 3 dense layer sampai dengan 5 dense layer. Hasil eksplorasi disajikan pada gambar 4 berikut:



Gambar 4: Akurasi dari setiap kemungkinan jumlah dense layer

Percobaan ini menghasilkan jumlah dense layer yang optimal sebanyak 2 dense layer dengan akurasi mencapai 92.82%.

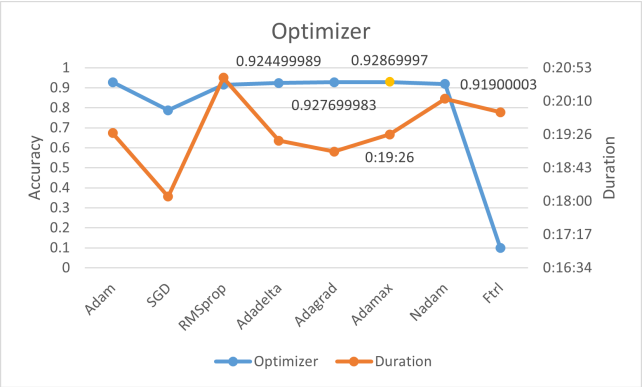
Percobaan berikutnya adalah mengecek jumlah neuron pada setiap dense layer yang optimal. Percobaan pertama dilakukan dengan menggunakan 24 neuron pada setiap dense layer. Percobaan berikutnya dilakukan dengan 32 neuron pada setiap dense layer (dengan kelipatan 2) sampai dengan 256 neuron pada setiap dense layer. Hasil eksplorasi disajikan pada gambar 5 berikut:

Accuracy						
		FC-2				
		24	32	64	128	256
FC-1	24	90.41%	91.07%	91.15%	91.60%	91.06%
	32	91.47%	91.93%	91.72%	92.00%	91.73%
	64	91.50%	91.71%	91.11%	91.85%	92.09%
	128	91.38%	92.10%	91.55%	**92.82%**	90.85%
	256	91.96%	92.27%	92.15%	92.47%	91.58%

Gambar 5: Akurasi dari setiap kemungkinan jumlah neuron pada setiap Dense Layer

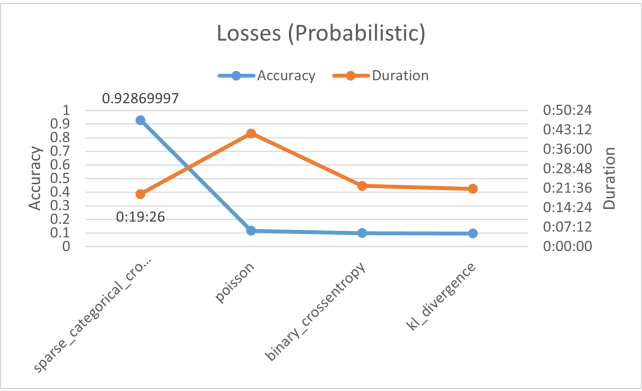
Dari percobaan tersebut, diperoleh akurasi mencapai 92.82% dengan jumlah neuron pada setiap dense layer sebanyak 128 neuron. Percobaan berikutnya adalah mengecek jenis optimizer yang disediakan oleh keras yang memberikan akurasi terbaik. Beberapa optimizer seperti Adam, SGD, RMSprop, Adagrad, Adadelta, Adamax, Nadam, TFOptimizer, dan FtrlOptimizer telah dieksplorasi. Hasil eksplorasi disajikan pada gambar 6 berikut:

Akurasi tertinggi diperoleh dengan menggunakan optimizer Adamax dengan



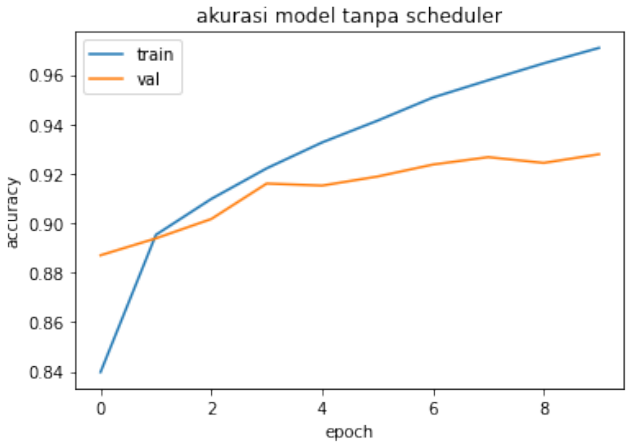
Gambar 6: Akurasi dari setiap optimizer yang disediakan oleh keras

akurasi mencapai 92.86%. Setelah mengecek optimizer optimal, langkah berikutnya adalah memilih loss function yang optimal. Loss function yang optimal adalah sparse categorical crossentropy. Hasil eksplorasi disajikan pada gambar 7 berikut:

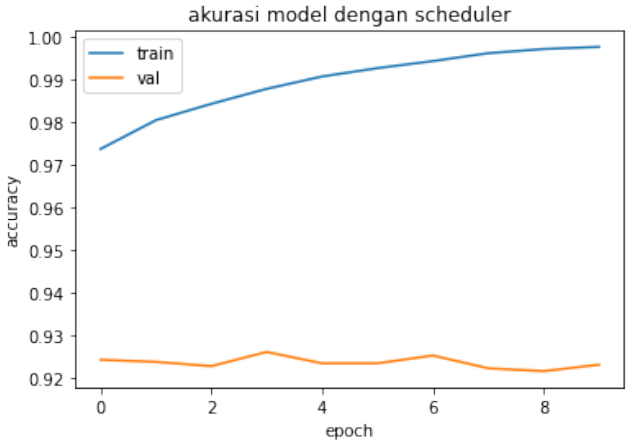


Gambar 7: Akurasi dari setiap loss function

Setelah deperoleh loss function optimal, langkah berikutnya adalah mengeksplorasi model dengan menambahkan learning rate scheduler. Percobaan dilakukan dengan mengkomparasi model tanpa scheduler dengan model dengan scheduler. Percobaan ini menggunakan 10 epoch dan learning rate sebesar 10^{-3}. Hasil tanpa menggunakan learning rate sceduler mengindikasikan bahwa akurasi pada training test cukup fluktuatif. Hasil eksplorasi tanpa learning rate scheduler disajikan pada gambar 8 dan pada gambar 9 disajikan hasil eksplorasi dengan learning rate scheduler.



Gambar 8: Akurasi dari setiap epoch tanpa learning rate scheduler



Gambar 9: Akurasi dari setiap epoch dengan learning rate scheduler

2.1 Kesimpulan

Hasil eksplorasi yang sudah dilakukan menunjukkan bahwa model yang optimal adalah model dengan jumlah kernel pada convolutional layer pertama sebanyak 32 dan jumlah kernel pada convolutional layer kedua sebanyak 128, jumlah dense layer sebanyak 2, jumlah neuron pada setiap dense layer sebanyak 128, optimizer Adamax, loss function sparse categorical crossentropy, dan learning rate scheduler.

3 Regresi

Persoalan regresi yang dilakukan menggunakan data Boston Housing yakni data yang berisi informasi lokasi, kedalaman tanah, dan harga rumah. Setiap catatan dalam dataset Boston Housing menggambarkan kota Boston. Data diambil dari Boston Standard Metropolitan Statistical Area (SMSA) pada tahun 1970. Atributnya didefinisikan sebagai berikut (diambil dari UCI Machine Learning Repository):

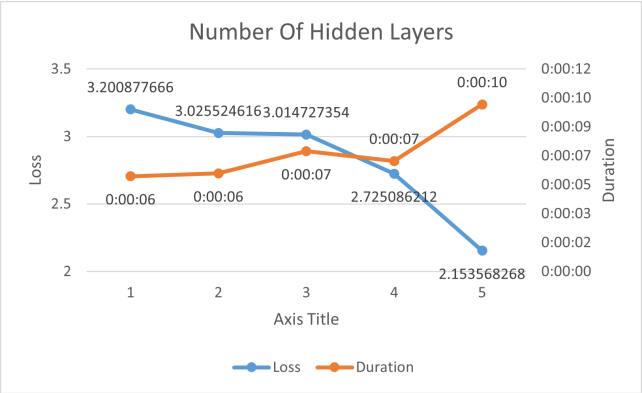
1. CRIM: tingkat kejahatan per kapita menurut kota
2. ZN: proporsi tanah perumahan yang dikategorikan untuk kavling lebih dari 25.000 sq.ft.
3. INDUS: proporsi hektar bisnis non-ritel per kota
4. CHAS: Variabel dummy Sungai Charles (= 1 jika saluran membatasi sungai; 0 sebaliknya)
5. NOX: konsentrasi oksida nitrat (bagian per 10 juta)
6. RM: rata-rata jumlah kamar per hunian
7. AGE: proporsi unit yang ditempati pemilik yang dibangun sebelum 1940
8. DIS: jarak tertimbang ke lima pusat kerja Boston
9. RAD: indeks aksesibilitas ke jalan raya radial
10. TAX: tarif pajak properti nilai penuh per $10.000
11. PTRATIO: rasio murid-guru menurut kota
12. B: 1000(Bk - 0.63)2 di mana Bk adalah proporsi orang kulit hitam menurut kota.
13. LSTAT: % status penduduk yang lebih rendah
14. MEDV: Nilai rata-rata rumah yang ditempati pemilik di $1000s per kamar.

Percobaan untuk forecasting nilai rata-rata rumah di kota Boston dilakukan dengan menggunakan neural network. Data yang digunakan adalah data training dan data testing. Data training dilakukan dengan menggunakan 80% data dan data testing dilakukan dengan menggunakan 20% data. Jumlah epoch yang digunakan sebanyak 200 epoch dan ukuran batch sebanyak 64.

Percebaan pertama dilakukan dengan mengecek jumlah dense layer yang optimal. Percobaan dilakukan dengan membangun model dengan 1 dense layer sampai dengan 5 dense layer. Hasil eksplorasi dari jumlah dense layer optimal adalah jumlah dense layer sebanyak 2. Loss yang dihasilkan sebesar 2.15 dengan menggunakan mean absolute error (MAE) dengan optimizer yang digunakan adalah RMSProp. Hasil eksplorasi dari jumlah dense layer optimal disajikan pada gambar 10.

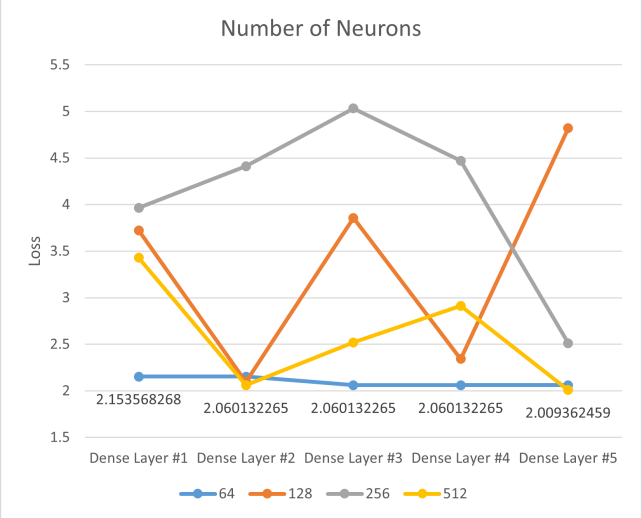
Berdasarkan hasil eksplorasi, jumlah dense layer optimal adalah 5 dense layer dengan durasi training sebanyak 200 epoch dan learning rate sebesar 10^{-3}.

Setelah mengecek jumlah dense layer optimal, langkah berikutnya adalah mengecek jumlah neuron optimal pada setiap layer. Percobaan pertama dilakukan dengan melakukan variasi jumlah neuron pada dense layer pertama. Jumlah neuron pada layer kedua sampai kelima dibiarkan tetap sebanyak 128. Percobaan berikutnya dilakukan pada dense layer kedua, ketiga sampai dengan



Gambar 10: Loss dan MAE dari setiap jumlah dense layer

dense layer kelima. Hasil eksplorasi dari jumlah neuron optimal disajikan pada gambar 11.



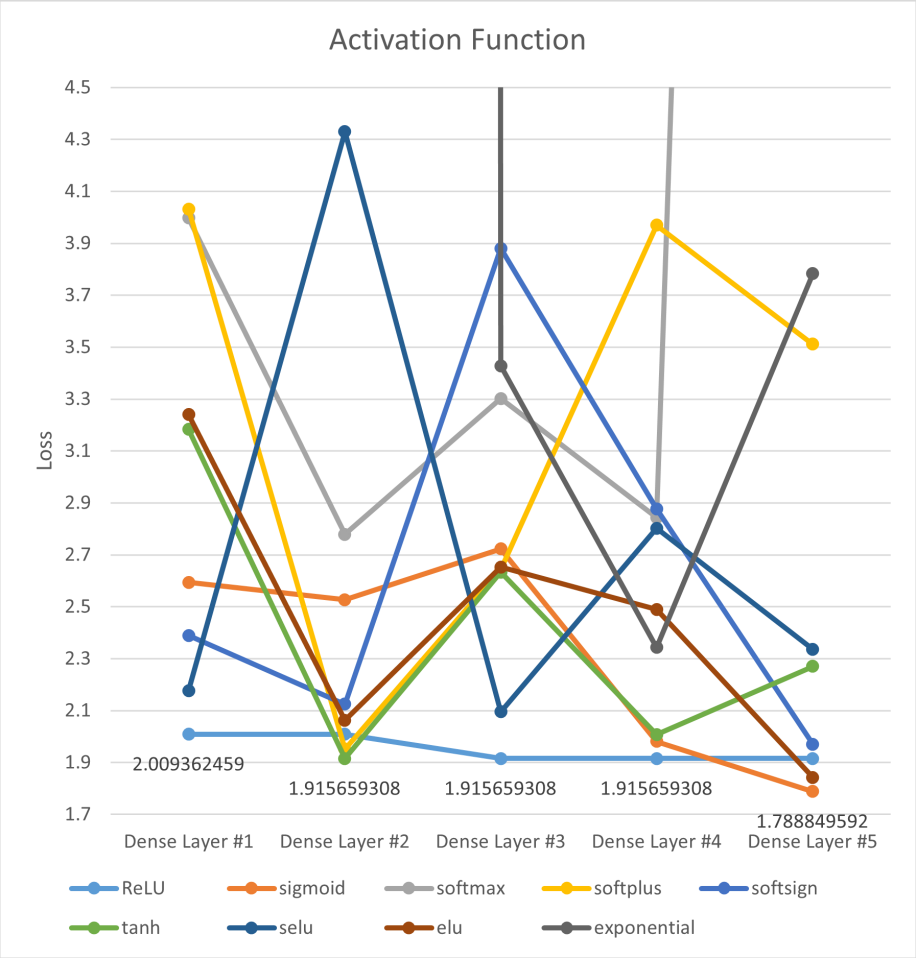
Gambar 11: Loss dari setiap percobaan variasi jumlah neuron pada setiap layer

Hasil eksplorasi yang diperoleh adalah sebagai berikut:

1. Jumlah neuron pada layer pertama optimal adalah 64 neuron.
2. Jumlah neuron pada layer kedua optimal adalah 512 neuron.
3. Jumlah neuron pada layer ketiga optimal adalah 64 neuron.
4. Jumlah neuron pada layer keempat optimal adalah 64 neuron.
5. Jumlah neuron pada layer kelima optimal adalah 512 neuron.

Setelah mengecek jumlah neuron untuk setiap dense layer, tahap berikut-

nya adalah mengecek activation function dari setiap dense layer yang optimal. Percobaan pertama dilakukan dengan melakukan variasi activation function pada layer pertama. Percobaan berikutnya dilakukan pada layer kedua, ketiga sampai dengan layer kelima. Hasil eksplorasi dari activation function optimal disajikan pada gambar 12.



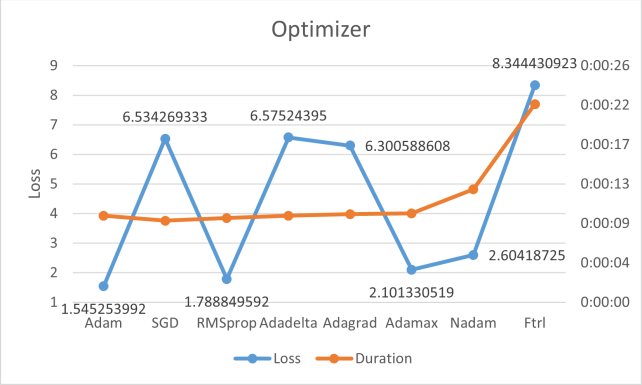
Gambar 12: Loss dari setiap percobaan variasi activation function pada setiap layer

Hasil eksplorasi yang diperoleh adalah sebagai berikut:

1. Activation function yang optimal pada layer pertama adalah ReLU.
2. Activation function yang optimal pada layer kedua adalah tanh.
3. Activation function yang optimal pada layer ketiga adalah ReLU.
4. Activation function yang optimal pada layer keempat adalah ReLU.

5. Activation function yang optimal pada layer kelima adalah sigmoid.

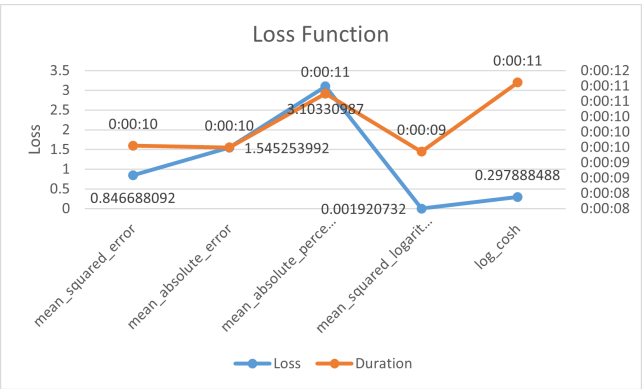
Tahap berikutnya adalah mengecek optimizer optimal yang digunakan pada model. Percobaan dilakukan dengan melakukan variasi optimizer pada model. Beberapa optimizer yang diujikan pada model ini adalah Adam, SGD, RMSprop, Adagrad, Adadelta, Adamax, Nadam, TFOptimizer, dan FtrlOptimizer. Hasil eksplorasi dari optimizer optimal disajikan pada gambar 13.



Gambar 13: Loss dari setiap percobaan variasi optimizer

Optimizer terbaik yang diperoleh adalah Adam dengan nilai loss sebesar 1.545.

Setelah mengecek optimizer optimal, tahap berikutnya adalah mengecek loss function optimal yang digunakan pada model. Percobaan dilakukan dengan melakukan variasi loss function pada model. Beberapa loss function yang diujikan pada model ini adalah mean square error (mse), mean absolute error (mae) dan lainnya. Hasil eksplorasi dari learning rate optimal disajikan pada gambar 14.



Gambar 14: Loss dari setiap percobaan variasi loss function

Loss funcion terbaik yang diperoleh adalah mean square logaritmic error dengan nilai loss sebesar 0.0019.

3.1 Kesimpulan

Kesimpulan dari eksplorasi yang dilakukan adalah sebagai berikut:

1. Jumlah dense layer yang optimal adalah 5 dense layer dengan jumlah neuron pada setiap dense layer adalah 64, 512, 64, 64, dan 512.
2. Activation function pada setiap dense layer yang optimal adalah ReLU, tanh, ReLU, ReLU, dan sigmoid.
3. Optimizer yang optimal adalah Adam dengan nilai loss sebesar 1.545.
4. Loss function yang optimal adalah mean square logaritmic error dengan nilai loss sebesar 0.0019.