

1. Revisi 4 (Bimbingan 1 April dengan Bu Herlina dan Bu Nurul)
 - Update abstrak: alasan urgensi membedakan dan mengetahui, buat ulang abstraknya [✓]
 - Cross Validation
 1. Tambahkan perhitungan cross validation nya di Bab 2 dan panggil dan Bab 3 (Coba lihat kang dylan atau kang ewen), letaknya di bagian evaluasi atau apa? [✓]
Tidak menggunakan Cross-validation, hanya sekedar membagi data train dan validation (data dibagi secara random).
 - Perbaiki Algoritma Adam nya
 1. Di 2.17 dan 2.18 $\beta_1^{(t)}$ atau β_1^t ini t nya iterasi atau apa? [✓]
Bukan iterasi tetapi pangkat sebesar nilai iterasinya
 2. Tambahkan proses backpropagation nya, itu penurunan Adam nya bagaimana di neural networknya hubungkan dengan CNN nya Kriteria berhenti Adam nya apa dan berapa? [✓]

Forward Propagation: Hitung Loss dan Cost

Backpropagation: Hitung gradien di setiap parameter (θ):

- Parameter weight dan bias di Dense layer
- Parameter weight dan bias di tiap layer Conv2D

Update dengan menggunakan Algoritma Adam

Catatan: saat $t = 0$ (inisiasi awal),

- weight dari kernel di **Conv2D** (sumber: [kernel_initializer="glorot_uniform"](#)) dicari dengan glorot uniform
- weight di layer **Dense** = 0
- moment = 0 ($m_0 = 0$)

- Perjelas lagi kaitan Adam di CNN nya
 1. Tambahkan penjelasan tentang stride [✓]
 2. Bagaimana proses akhir forward propagation? Bagaimana kaitan antara Softmax dan Categorical cross-entropy? [✓]
 - Perjelas apa itu cross-entropy? [✓] di 2.8
 - Berapa banyak jenis cross-entropy? [✓] 2: Categorical dan Binary
 - Apa itu aktivasi softmax? [✓]
 - Bagaimana hubungan softmax dengan CC? [✓]

Rumus di 2.8 Pengoptimal poin 1&2 dan ,
pemakaian di 3.37 – 3.41

3. Bagaimana proses update parameter (weight dan bias) dilakukan?

- Apakah berjenjang tiap layer diurutkan dari belakang? (backpropagation) ataukah langsung seluruh layer? [✓]

Ya, berjenjang per layer, tetapi karena parameter nya sangat banyak jadi hanya disebutkan cari gradien cost function (persamaan 2.20) terhadap parameter nya

- Apakah update weight dilakukan setelah satu batch data selesai diklasifikasi pada proses forward prop atau langsung per epoch? [✓] setiap selesai train satu mini-batch, baru hitung cost function (persamaan 2.20), hitung gradien parameternya (persamaan 2.21), terus update setiap parameternya dengan Adam (persamaan 2.22 – 2.26)

- Apakah perlu ditambahkan penjelasan mengenai batch_size di bagian image prep? [✓] ditambah di halaman 9-10
- Di bab 2, semua package dan fungsi seperti Conv2D Pillow begitu disatukan di sub bab Python, sub bab matplotlib dan tensorflow juga dimasukkan jadi sub bab dari sub bab Python. [✓]

Catatan Tambahan: Bebera keterangan tentang library nya tetap dimation di bagian terkait contohnya di 2.4 karena mengandung teori yang digunakan pada proses konvolusinya, tambahan: tidak dibahas secara detail hingga mencantumkan gambar seluruh parameter Conv2D

```
tf.keras.layers.Conv2D(  
    filters,  
    kernel_size,  
    strides=(1, 1),  
    padding='valid',  
    data_format=None,  
    dilation_rate=(1, 1),  
    groups=1,  
    activation=None,  
    use_bias=True,  
    kernel_initializer='glorot_uniform',  
    bias_initializer='zeros',  
    kernel_regularizer=None,  
    bias_regularizer=None,  
    activity_regularizer=None,  
    kernel_constraint=None,  
    bias_constraint=None,  
    **kwargs  
)
```

Gambar 2.3 – Hyperparameter dalam fungsi Conv2D
(TensorFlow Developer, 2024a)

- 2.8 dan 2.9 variabel i dan k artinya sama sehingga i saja [✓]
- Bab 1.3 Batasan masalah tambah tentang i di *learning rate* [✓]
- Coba untuk testing hasilnya berapa? [✓] 81,9%

Tambahan:

- Penyeragaman notasi operasi konvolusi **dari**

$$y_{i,j,k} = \sum_{a=1}^{k_1-1} \sum_{b=1}^{k_2-1} \sum_{c=1}^{k_3-1} I(i+a, j+b, k+c) * K(a, b, c)$$

menjadi:

$$y_{i,j,k} = \sum_{a=1}^{k_1-1} \sum_{b=1}^{k_2-1} \sum_{c=1}^{k_3-1} X(i+a, j+b, k+c) * W(a, b, c)$$