1. [Nilai: 25] Diberikan model ensemble dengan 3 base classifiers untuk klasifikasi biner (kelas Pos atau Neg). Setiap base classifier memberikan prediksi dalam bentuk peluang kelas Pos. Ketika diberikan sebuah instance **x**, berikut hasil prediksinya:

| Base-classifier | Bobot | Prediksi |
| --- | --- | --- |
| Model 1 | 0.25 | 0.2 |
| Model 2 | 0.45 | 0.75 |
| Model 3 | 0.3 | 0.4 |

Lakukanlah inferensi model ensemble untuk mendapatkan hasil prediksi akhir apakah **x** termasuk dalam kelas Pos atau Neg dengan skema berikut. *Sebelum menjawab kelas prediksinya, berikanlah terlebih dahulu metode inferensi dari setiap skema*.

* 1. Bagging
  2. Random Forest
  3. Heterogeneous ensemble with weighting
  4. Heterogeneous ensemble with meta-learning, dengan level 1 base-estimatorsnya adalah model 1 dan 2, dan level 2 estimator adalah model 3.
  5. AdaBoost

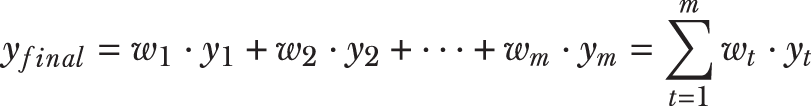
Catatan:

* Jawaban prediksi tanpa penjelasan caranya tidak akan dinilai.
* Jika terdapat beberapa versi algoritma yg dipelajari di kuliah, sebutkan versi mana yang digunakan.
* Jika terdapat bagian yang tidak jelas, tuliskanlah asumsinya secara eksplisit.

1. [Nilai 5] Jelaskanlah proses training dan inferensi pada Gradient Boosting.

Jawab:

| Base-classifier | Bobot | Prediksi | **Kelas** |
| --- | --- | --- | --- |
| Model 1 | 0.25 | 0.2 | **Neg** |
| Model 2 | 0.45 | 0.75 | **Pos** |
| Model 3 | 0.3 | 0.4 | **Neg** |

1. Bagging: inferensi dengan voting, return majority vote, yaitu   
   argmax{pos: 1, neg: 2} = Neg
2. RF: inferensi dgn mean predicted class probabilities, yaitu (0.2+0.75+0.4)/3=0.45 ⇒ Neg. Untuk kuis 1 ini, jawaban dgn majority vote masih dianggap benar.
3. Voting: inferensi dgn kombinasi linear, yaitu: 0.25\*0.2+0.45\*0.75+0.3\*0.4=0.5075 ⇒ Pos
4. Stacking: inferensi dari model 3, yaitu: Neg
5. Adaboost versi Han: inferensi dgn menghitung bobot setiap kelas, lalu return kelas dgn peluang tertinggi. WPos: 0.45; wNeg: 0.25+0.3=0.55 ⇒ Neg

Adaboost versi Kunapuli: sigma alfa\*h = 0.25\*0.2+0.45\*0.75+0.3\*0.4=0.5075 ⇒ Pos

2. Training:

* Model iterasi awal dan residual sebagai target prediksi iterasi berikutnya
* Weak learners

Inferensi:

* Inferensi base-model masing-masing
* Integrasi final prediction dgn kombinasi linear

**Bagian II**

Terdapat dataset berlabel yang terdiri atas 3 instances dengan 4 fitur/atribut (f1 s.d. f4) berikut ini, untuk klasifikasi sentimen terhadap data review produk. Pembelajaran untuk membentuk hipotesis (model) dilakukan dengan Perceptron - Batch Gradient Descent (fungsi aktivasi linear). Encoding kelas positif adalah 1 dan kelas negatif adalah -1, dengan batas untuk kelas positif > 0. Inisialisasi bobot semua fitur adalah 1, dan inisialisasi bobot bias adalah -1. Learning rate 0.1, Error threshold 0.5, Max epoch: 5.

Data training:

| **Review** | **f1** | **f2** | **f3** | **f4** | **sentimen** |
| --- | --- | --- | --- | --- | --- |
| r1 | 1 | 0 | 0 | 1 | 1 |
| r2 | 0 | 0 | 1 | 0 | -1 |
| r3 | 1 | 1 | 0 | 1 | 1 |

1. Lakukan proses pembelajaran model dengan menuliskan detil perhitungan di setiap epoch. Tuliskan bobot yang dihasilkan dan error dari hasil pembelajaran. [Nilai: 25]
2. Diberikan data uji sebagai berikut, lakukan prediksi kelas sentimen dengan menggunakan model yang dihasilkan pada bagian a). Hitung akurasi model. [Nilai: 10]

| **Review** | **f1** | **f2** | **f3** | **f4** | **sentimen** |
| --- | --- | --- | --- | --- | --- |
| r4 | 0 | 0 | 0 | 0 | -1 |
| r5 | 1 | 1 | 0 | 0 | 1 |

Solusi:

1. Tabel pembelajaran

|  | w\_i0 | w\_i1 | w\_i2 | w\_i3 | w\_i4 | x0 | f1 | f2 | f3 | f4 | Target | Output | Target – Output | Delta w\_i0 | Delta w\_i1 | Delta w\_i2 | Delta w\_i3 | Delta w\_i4 | Error |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | -1 | 1 | 1 | 1 | 1 |  |  |  |  |  |  |  | Bobot awal | 0 | 0 | 0 | 0 | 0 | 0 |
| Epoch 1 | -1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | -1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | -1 | 0 | -1 | -0.1 | 0 | 0 | -0.1 | 0 | 0.5 |
|  | -1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 2 | -1 | -0.2 | -0.1 | -0.1 | -0.1 | -0.1 | 1 |
|  |  |  |  |  |  |  |  |  |  |  |  |  | Total | -0.3 | -0.1 | -0.1 | -0.2 | -0.1 | 1.5 |
| Update bobot | -1.3 | 0.9 | 0.9 | 0.8 | 0.9 |  |  |  |  |  |  |  | Bobot awal | 0 | 0 | 0 | 0 | 0 | 0 |
| Epoch 2 | -1.3 | 0.9 | 0.9 | 0.8 | 0.9 | 1 | 1 | 0 | 0 | 1 | 1 | 0.5 | 0.5 | 0.05 | 0.05 | 0 | 0 | 0.05 | 0.125 |
|  | -1.3 | 0.9 | 0.9 | 0.8 | 0.9 | 1 | 0 | 0 | 1 | 0 | -1 | -0.5 | -0.5 | 0 | 0.05 | 0 | -0.05 | 0.05 | 0.25 |
|  | -1.3 | 0.9 | 0.9 | 0.8 | 0.9 | 1 | 1 | 1 | 0 | 1 | 1 | 1.4 | -0.4 | -0.04 | 0.01 | -0.04 | -0.05 | 0.01 | 0.33 |
|  |  |  |  |  |  |  |  |  |  |  |  |  | Total | 0.01 | 0.11 | -0.04 | -0.1 | 0.11 | 0.705 |
| Update bobot | -1.29 | 1.01 | 0.86 | 0.7 | 1.01 |  |  |  |  |  |  |  | Bobot awal | 0 | 0 | 0 | 0 | 0 | 0 |
| Epoch 3 | -1.29 | 1.01 | 0.86 | 0.7 | 1.01 | 1 | 1 | 0 | 0 | 1 | 1 | 0.73 | 0.27 | 0.027 | 0.027 | 0 | 0 | 0.027 | 0.03645 |
|  | -1.29 | 1.01 | 0.86 | 0.7 | 1.01 | 1 | 0 | 0 | 1 | 0 | -1 | -0.59 | -0.41 | -0.014 | 0.027 | 0 | -0.041 | 0.027 | 0.1205 |
|  | -1.29 | 1.01 | 0.86 | 0.7 | 1.01 | 1 | 1 | 1 | 0 | 1 | 1 | 1.59 | -0.59 | -0.073 | -0.032 | -0.059 | -0.041 | -0.032 | 0.29455 |
|  |  |  |  |  |  |  |  |  |  |  |  |  | Total | -0.06 | 0.022 | -0.059 | -0.082 | 0.022 | 0.4515 |
| Bobot akhir | -1.35 | 1.032 | 0.801 | 0.618 | 1.032 |  |  |  |  |  |  |  |  |  |  |  |  |  | Error < threshold : stop |

1. Prediksi Data baru

| **Review** | **f1** | **f2** | **f3** | **f4** | **sentimen** | **Output** | **Kelas** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| r4 | 0 | 0 | 0 | 0 | -1 | -1.35 | -1 |
| r5 | 1 | 1 | 0 | 0 | 1 | 0.483 | 0 |

Akurasi = ½ = 50%

**Bagian III**

(Bagian III terdiri atas 6 soal)

Terdapat sebuah arsitektur Feed Forward Neural Network (FFNN) sebagai berikut, untuk mempelajari dataset yang memiliki 3 fitur input dengan sebuah label kelas. Arsitektur FFNN memiliki dua *hidden layer*, dan tiap *hidden layer* memiliki dua neuron; serta sebuah neuron pada *output layer*. Aturan penamaan yang digunakan adalah sebagai berikut:

(i) input adalah xi, dengan i terurut mulai dari 1

(ii) hidden layer pertama h1-i, dengan i terurut mulai dari 1

(iii) hidden layer kedua h2-i, dengan i terurut mulai dari 1

(iv) prediksi output adalah oi, dan target output adalah ti, dengan i terurut mulai dari 1

(v) bobot adalah w[m][n], dengan m adalah asal dan n adalah tujuan

(vi) jika diperlukan variabel tambahan, silahkan diberikan penamaan sendiri secara konsisten.

Jawablah pertanyaan (1 - 5) berikut pada tempat yang disediakan berdasarkan spesifikasi di atas.

| Nomor | Pertanyaan | Jawaban |
| --- | --- | --- |
| 1 | Gambarkan arsitektur FFNN secara eksplisit sesuai spesifikasi, lengkap untuk semua variabel dan bobot yang diperlukan, beserta penamaannya.  (Nilai 6)  Catatan: bias selalu ada (ingat perhitungan banyaknya parameter yang perlu dipelajari di FFNN). |  |
| 2 | Tentukan banyaknya parameter yang perlu dipelajari berdasarkan spesifikasi dan arsitektur yang telah digambarkan. Tuliskan angka yang terlibat dan tidak hanya hasil akhir saja. (Nilai 6) | [(3+1) \* 2] + [(2+1) \* 2] + [(2+1) \* 1] = 17 |
| 3 | Jika hidden layer pertama menggunakan fungsi aktivasi sigmoid, dan hidden layer kedua menggunakan fungsi aktivasi ReLU, serta output layer menggunakan fungsi aktivasi linear, tentukan output h2-1. Tuliskan dalam bentuk formula detil dengan variabel-variabel yang terlibat, mulai dari input. Fungsi aktivasi tidak dituliskan nama fungsi saja, tapi lengkap formulanya.  (Nilai 6) | net\_h1-1 = x1 \* w[x1][h1-1] + x2 \* w[x2][h1-1] + x3 \* w[x3][h1-1] + w[b1][h1-1]  h1-1 = 1/ (1+ exp(- net\_h1-1)  net\_h1-2 = x1 \* w[x1][h1-2] + x2 \* w[x2][h1-2] + x3 \* w[x3][h1-2] + w[b1][h1-2]  h1-2 = 1/ (1+ exp(- net\_h1-2)  net\_h2-1 = h1-1 \* w[h1-1][h2-1] + h1-2 \* w[h1-2][h2-1] + w[b2][h2-1]  **h2-1 = max{0,net\_h2-1}** |
| 4 | Jika: (i) fungsi *loss/ cos*t (E) yang digunakan adalah Squarred Error dengan turunan terhadap output adalah:  (-1) \* (ti - oi);  (ii) turunan fungsi linear adalah: 1;  (iii) turunan fungsi ReLU adalah: 1 jika input fungsi > 0, dan 0 jika input fungsi ≤ 0;  (iv) turunan fungsi sigmoid adalah: output\*(1 - output);  (v) error term dari o1 adalah 𝛅o1 (𝛅o1 = ) ;  Tentukan (dalam formula rinci) nilai 𝛅h2-1 (error term untuk h2-1, 𝛅h2-1 = (Nilai 6) | Error term: 𝛅h2-1 =    = \* \*    = (-1) \* (ti - oi) \* w[h2-1][o1] \* (1 if net\_h2-1>0, 0 otherwise) |
| 5 | Tentukan nilai update untuk w[h2-1][o1] dalam bentuk formula yang melibatkan variabel-variabel yang diperlukan, jika learning rate adalah 𝜼.  (Nilai 6) | w[h2-1][o1] = w[h2-1][o1] - 𝜼 \*  = w[h2-1][o1] - 𝜼 \*((-1)\*(ti - oi)\*1\*h[2-1] |

6. Lingkarilah pernyataan yang benar dari lima pernyataan berikut ini. (Nilai 5)

1. Semakin ~~dalam~~ lebar suatu FFNN, artinya semakin banyak neuron yang digunakan dalam satu *hidden layer*.
2. Penggunaan nilai *learning rate* yang semakin besar ~~menjamin pencapaian~~ *~~global cost (lost) minimum~~* ~~semakin cepat~~, sehingga meminimalkan jumlah epoch.
3. Nilai hasil fungsi aktivasi sigmoid tidak pernah bernilai negatif.



1. Jika semua bobot dalam FFNN bernilai 0.1, dan setiap hidden layer hanya berisi 1 neuron serta menggunakan fungsi aktivasi sigmoid untuk seluruh layer, maka saat dilakukan Backpropagation, FFNN dengan 10 hidden layer akan menghasilkan nilai gradient lebih kecil di hidden layer pertama, dibandingkan dengan FFNN dengan 5 hidden layer.
2. ~~Bobot~~ dari bias selalu bernilai 1.