

PENGENALAN ANGKA TULISAN TANGAN MENGGUNAKAN JARINGAN SYARAF BUATAN

Yohanes Bowo Widodo¹, M. Amin Sakaria²

^{1,2}Universitas Mohammad Husni Thamrin

ybowowidodo@gmail.com

ABSTRAK

Latar belakang: Adalah suatu tantangan untuk membuat program yang berperilaku cerdas seperti manusia. Khususnya bagaimana membuat program yang dapat belajar untuk menyelesaikan masalah tertentu. Pada penelitian ini, dikembangkan suatu program komputer yang menerapkan jaringan syaraf tiruan yang dapat mempelajari bagaimana mengenali angka tulisan tangan. Fokus pada pengenalan tulisan tangan ini dipilih karena merupakan prototipe masalah yang sangat baik untuk mempelajari bagaimana perilaku jaringan syaraf secara umum.

Tujuan: Membuat program cerdas yang dapat mengenali angka tulisan tangan.

Metode: Jaringan syaraf buatan dengan arsitektur backpropagation dan algoritma pembelajaran stochastic gradient descent yang diimplementasikan dengan bahasa pemrograman Python.

Hasil: Program dapat mengenali angka tulisan tangan dengan ketepatan lebih dari 96 persen, tanpa intervensi manusia.

Kesimpulan dan Saran: Algoritma pembelajaran yang baik, apabila dilatih dengan data pelatihan yang kurang baik, maka unjuk kerjanya akan lebih buruk daripada algoritma sederhana yang dilatih dengan data pelatihan yang baik. Disarankan agar penelitian ini dapat dikembangkan untuk masalah tulisan tangan yang lebih luas, yaitu selain mengenali angka juga dapat mengenali huruf tulisan tangan.

Kata Kunci : pengenalan tulisan tangan, jaringan syaraf tiruan, algoritma pembelajaran, Perambatan Balik, Stochastic Gradient Descent

ABSTRACT

Background: It is a challenge to build a program that behave intelligent like human. Specially how to build a program that able to learn in order to solve a specific problem. In this reserarch will be developed a computer program implementing an artificial neural network that learns to recognize handwritten digits. The focus on handwriting recognition because it is an excellent prototype problem for learning about neural network in general.

Aims: Building an intelligent program that able to recognize hand written number (digits).

Method: Artificial Neural Network with Backpropagation architecture and Stochastic gradient descent learning algorithm wich is implemented in Python programming language.

Result: The program can recognize handwritten digits with an accuracy over 96 percent, without human intervention.

Conclusion and Advice: A good learning algorithm that be learned with bad learning data will perform worse than a simple learning algorithm that be learned with good learning data. It is suggested that this research to be improved with capability to recognize handwritten letter.

Key Words : handwritten recognition, artificial neural network, learning algorithm, Backpropagation, Stochastic Gradient Descent

PENDAHULUAN

Sistem penglihatan manusia adalah suatu yang amat hebat. Misalnya dalam mengenali angka tulisan tangan berikut :



Kebanyakan orang tanpa kesulitan dapat mengenali angka tersebut sebagai 5 0 4 1 9 2. Dalam hal pengenalan tulisan tangan, kecerdasan manusia sangat mengagumkan. Oleh karena itu salah satu bidang dalam kecerdasan buatan adalah bagaimana membuat komputer mampu menirukan kecerdasan manusia, misalnya dalam mengenali tulisan tangan.

Kesulitan dari pengenalan pola visual muncul dalam pembuatan program komputer yang dapat mengenali angka seperti tersebut diatas. Apa yang mudah bagi manusia menjadi sesuatu yang sangat sukar bagi program komputer. Misalnya untuk mengenali angka 9, aturannya adalah “memiliki bulatan diatas, dan garis vertikal di kanan, dilanjutkan ke arah kiri”. Hal tersebut sangat sulit dinyatakan secara algoritmik. Ketika dicoba untuk membuat aturan tersebut secara pasti, akan terdapat banyak pengecualian dan kasus khusus yang harus ditangani.

Jaringan syaraf buatan memiliki pendekatan yang berbeda terhadap masalah tersebut. Idennya adalah mengambil sejumlah besar angka tulisan tangan tersebut sebagai contoh pelatihan, dan mengembangkan sistem yang dapat belajar dari contoh pelatihan tersebut. Dengan perkataan lain, Jaringan syaraf menggunakan contoh untuk secara otomatis menemukan aturan untuk mengenali angka tulisan tangan.



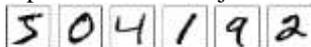
Jaringan syaraf tidak belajar dari aturan mengenai bentuk tulisan tangan, melainkan dari contoh tulisan tangan yang dimasukkan sebagai pola pelatihan. Dengan meningkatnya jumlah contoh pelatihan, jaringan dapat belajar mengenali tulisan tangan dan dapat meningkatkan ketepatannya. Kemungkinan untuk membangun jaringan syaraf yang baik, diperlukan ratusan bahkan puluhan ribu contoh masukan.

METODE

Permasalahan pengenalan angka tulisan tangan dapat dibagi dalam dua sub masalah. Pertama disebut masalah segmentasi. Disini harus dicari cara untuk memisahkan gambar yang terdiri dari beberapa angka menjadi deretan gambar yang terpisah, yang masing-masing terdiri dari satu angka. Misalnya gambar deretan angka



dipisahkan menjadi enam gambar terpisah



Manusia dapat menyelesaikan masalah segmentasi ini dengan mudah. Namun menjadi masalah yang cukup menantang untuk membuat program komputer yang dapat secara tepat memisahkan gambar tersebut.

Setelah gambar disegmentasi, maka sub masalah yang kedua, program harus mengklasifikasi setiap angka

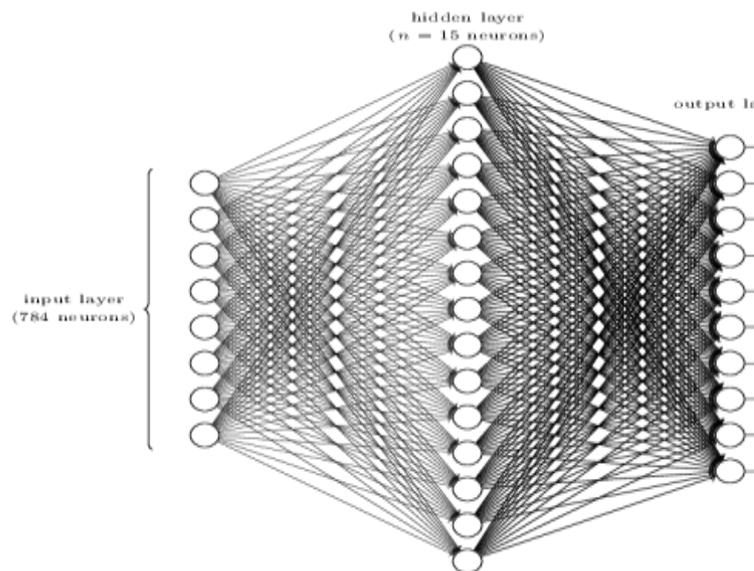
yang telah terpisah tersebut. Misalnya harus mengenali tulisan angka pertama



Sebagai angka 5.

Penelitian ini akan difokuskan pada permasalahan yang kedua, yaitu pengklasifikasian angka tunggal. Hal ini dilakukan karena masalah segmentasi tidak terlalu sulit untuk dipecahkan.

Untuk mengenali suatu angka tertentu digunakan jaringan syaraf tiga lapisan,



Lapisan input dari jaringan berisi neuron-neuron yang menerima nilai dari piksel input. Data pelatihan untuk jaringan terdiri dari gambar 28 x 28 piksel dari angka tulisan tangan yang telah dipindai, sehingga lapisan input terdiri dari 28 x 28 neuron = 784. Untuk menyederhanakan, pada gambar hanya digambarkan 8 neuron.

Piksel input adalah tingkat keabuan (grey scale), dengan nilai 0 menyatakan putih, nilai 1 menyatakan hitam, dan nilai di antara keduanya menyatakan tingkat keabuan yang menaik secara bertahap.

Lapisan kedua dari jaringan adalah lapisan tersembunyi (hidden layer). Jumlah neuron dalam lapisan tersembunyi dinyatakan dengan n_j dan akan di uji coba beberapa nilai n_j yang berbeda. Pada gambar diperlihatkan lapisan tersembunyi dalam jumlah sedikit, hanya terdiri dari $n = 15$ neuron.

Lapisan output dari jaringan terdiri dari 10 neuron. Jika neuron pertama aktif, maka output ≈ 0 . Hal itu mengindikasikan bahwa jaringan menganggap bahwa angka tersebut adalah 0. Dan seterusnya, secara lebih tepat, neuron keluaran diberi nomor antara 0 sampai 9, dan akan ditentukan neuron mana yang memiliki nilai aktivasi yang paling tinggi. Jika neuron tersebut, katakan neuron nomor 6, maka jaringan akan memperkirakan bahwa angka masukan adalah 6, dan seterusnya untuk neuron keluaran yang lain.

Jaringan syaraf buatan yang digunakan berarsitektur Backpropagation, dengan algoritma pembelajaran standar untuk jaringan syaraf yaitu stochastic gradient descent. Program diimplementasikan menggunakan bahasa pemrograman Python 2.7. Dalam penelitian digunakan gambar untuk pelatihan sebanyak 50.000 data untuk melatih jaringan syaraf, dan gambar untuk menguji jaringan syaraf sebanyak 10.000 data gambar.

Program inti dari jaringan syaraf adalah program untuk inialisasi objek jaringan syaraf :

```
class Network(object):
```

```
    def __init__(self, sizes):
        self.num_layers = len(sizes)
        self.sizes = sizes
        self.biases = [np.random.randn(y, 1) for y in
            sizes[1:]]
        self.weights = [np.random.randn(y, x)
            for x, y in zip(sizes[:-1], sizes[1:])]
```

Pusatnya pada kelas Network yang digunakan untuk merepresentasikan jaringan syaraf. Pada bagian program tersebut, list sizes berisi jumlah neuron pada jaringan syaraf secara berturutan. Contohnya, jika ingin membuat jaringan dengan 784 neuron di lapisan pertama, 300 neuron di lapisan kedua, dan 10 neuron di lapisan terakhir, kita lakukan dengan cara berikut :

```
net = Network([784, 300, 10])
```

Bias dan bobot dari objek jaringan di inialisasi secara acak menggunakan fungsi

```
np.random.randn
```

Nilai acak ini digunakan oleh algoritma stochastic gradient descent sebagai nilai awal sebelum dilakukan pembelajaran.

Program menghitung keluaran dari suatu jaringan dimulai dengan mendefinisikan fungsi sigmoid :

```
def sigmoid(z):
    return 1.0/(1.0+np.exp(-z))
```

jika z adalah vektor atau array, maka fungsi sigmoid diterapkan pada setiap elemen, yaitu dalam bentuk tervektorisasi. Kemudian ditambahkan method feedforward pada kelas Network untuk setiap lapisan,

```
def feedforward(self, a):
    """Menghitung keluaran jaringan jika masukannya
    adalah "a"."""
    for b, w in zip(self.biases, self.weights):
        a = sigmoid(np.dot(w, a)+b)
    return a
```

Hal utama yang harus dilakukan objek Network adalah belajar. Untuk itu dibuat method SGD yang mengimplementasikan stochastic gradient descent sebagai berikut:

```
def SGD(self, training_data, epochs, mini_batch_size, eta,
        test_data=None):
    """Melatih jaringan syaraf menggunakan mini-batch
    stochastic gradient descent. "training_data" adalah list dari
    tuples "(x, y)" menyatakan input pelatihan dan keluaran
    yang dikehendaki.
    Parameter opsional lainnya adalah sesuai
    dengan arti Namanya.
    Jika "test_data" disediakan, maka jaringan
    akan dievaluasi terhadap
    data uji setelah selesai setiap epoch, dan
    perkembangan sementara dicetak. Hal ini
    berguna untuk melacak kemajuan proses,
    namun sedikit memperlambat keseluruhan
    proses."""
    if test_data: n_test = len(test_data)
    n = len(training_data)
    for j in xrange(epochs):
        random.shuffle(training_data)
        mini_batches = [
            training_data[k:k+mini_batch_size]
            for k in xrange(0, n, mini_batch_size)]
        for mini_batch in mini_batches:
            self.update_mini_batch(mini_batch, eta)
        if test_data:
            print "Epoch {0}: {1} / {2}".format(
                j, self.evaluate(test_data), n_test)
        else:
            print "Epoch {0} complete".format(j)
```

HASIL DAN DISKUSI

Pesan keluaran berikut menunjukkan jumlah gambar contoh yang telah dikenali dengan benar oleh jaringan setelah setiap epoch pelatihan. Sebagaimana terlihat, pada epoch pertama pengenalan mencapai 9.129

dari 10.000 gambar, dan terus meningkat pada epoch berikutnya.

Epoch 0: 9129 / 10000

Epoch 1: 9295 / 10000

Epoch 2: 9348 / 10000

...

Epoch 27: 9528 / 10000

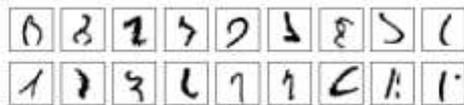
Epoch 28: 9542 / 10000

Epoch 29: 9534 / 10000

Jaringan yang dilatih memberikan tingkat ketepatan sekitar 95 persen pada saat puncak (Epoch 28), suatu hal yang sangat menggembirakan.

Jika jumlah lapisan tersembunyi diubah menjadi 100 neuron, ketepatan meningkat menjadi 96.59 persen. Dalam kasus ini, penambahan jumlah neuron lapisan tersembunyi dapat meningkatkan hasil menjadi lebih baik.

Kemampuan jaringan syaraf yang telah dilatih dapat menyamai kemampuan manusia, bahkan dapat dikatakan lebih baik, karena data yang di ujicobakan termasuk beberapa data yang sangat sulit dikenali bahkan oleh manusia, misalnya bentuk berikut :



Bentuk gambar tersebut sangat sulit diklasifikasikan sehingga dari 10.000 data yang dicobakan, terdapat 341 gambar yang tidak bisa dikenali, yaitu gambar seperti contoh diatas.

KESIMPULAN DAN REKOMENDASI

Program pengenalan angka tulisan tangan menggunakan jaringan syaraf buatan yang telah dikembangkan dengan bahasa pemrograman Python mampu mengenali 96 persen contoh gambar pengujian yang berjumlah 10.000 gambar, sehingga bisa disimpulkan bahwa kemampuannya menyamai bahkan melebihi kecerdasan manusia dalam mengenali angka tulisan tangan.

Dalam beberapa penelitian yang lain dapat diambil kesimpulan bahwa

Algoritma Canggih \leq Algoritma Sederhana + Data pelatihan yang baik.

Jadi, algoritma pembelajaran yang baik, apabila dilatih dengan data pelatihan yang kurang baik, maka unjuk kerjanya akan lebih buruk daripada algoritma sederhana yang dilatih dengan data pelatihan yang baik.

Sebagai rekomendasi, disarankan agar penelitian ini dapat dikembangkan untuk masalah tulisan tangan yang lebih luas, yaitu selain mengenali angka juga dapat mengenali huruf tulisan tangan.

DAFTAR PUSTAKA

Desiani, A. & Arhami, M. (2007) Konsep Kecerdasan Buatan. Yogyakarta, Penerbit Andi.

Kusumadewi, S. (2003) Artificial Intelligence, Teknik dan Aplikasinya. Yogyakarta, Graha Ilmu.

Sutojo, T. & Mulyanto, E. & Suhartono, V. (2011) Kecerdasan Buatan. Yogyakarta, Penerbit Andi.

<http://neuralnetworksanddeeplearning.com/chap1.html>