PERTEMUAN 12

Konsep Deep Learning

DL memberikan hasil terobosan dalam pengenalan suara dan klasifikasi gambar

From this Hinton et al 2012 paper:
http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/38131.pdf

modeling technique	#params [10^6]	WER Hub5'00-SWB	WER RT03S-FSH
GMM, 40 mix DT 309h SI	29.4	23.6	27.4
NN 1 hidden-layer×4634 units	43.6	26.0	29.4
+ 2×5 neighboring frames	45.1	22.4	25.7
DBN-DNN 7 hidden layers×2048 units	45.1	17.1	19.6
+ updated state alignment	45.1	16.4	18.6
+ sparsification	15.2 nz	16.1	18.5
GMM 72 mix DT 2000h SA	102.4	17.1	18.6

task	hours of training data	DNN-HMM	GMM-HMM with same data	GMM-HMM with more data
Switchboard (test set 1)	309	18.5	27.4	18.6 (2000 hrs)
Switchboard (test set 2)	309	16.1	23.6	17.1 (2000 hrs)
English Broadcast News	50	17.5	18.8	
Bing Voice Search (Sentence error rates)	24	30.4	36.2	
Google Voice Input	5,870	12.3		16.0 (>>5,870hrs)
Youtube	1,400	47.6	52.3	

go here: http://yann.lecun.com/exdb/mnist/

From here:
http://people.idsia.ch/~juergen/cvpr2012.pdf

Dataset	Best result of others [%]	MCDNN [%]	Relative improv. [%]
MNIST	0.39	0.23	41
NIST SD 19	see Table 4	see Table 4	30-80
HWDB1.0 on.	7.61	5.61	26
HWDB1.0 off.	10.01	6.5	35
CIFAR10	18.50	11.21	39
traffic signs	1.69	0.54	72
NORB	5.00	2.70	46

Apa sebenarnya ***deep Learning***?
mengapa umumnya lebih baik daripada metode lain pada gambar (*IMAGE*), ucapan (*Speech*) dan jenis data tertentu lainnya?

The short answers

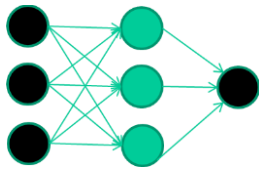
1. **'Deep Learning' berarti** menggunakan jaringan syaraf **(neural network)**
   dengan **beberapa lapisan node** antara input dan output

2. **S**erangkaian lapisan antara input & output lakukan identifikasi dan pemrosesan fitur dalam serangkaian tahapan**,** seperti otak kita tampaknya.

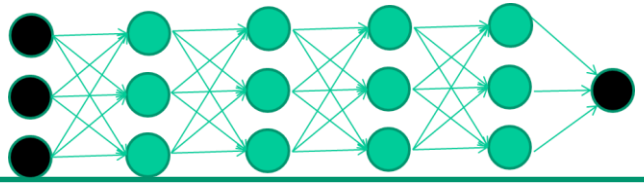
jaringan saraf multilayer telah ada selama 25 tahun.

Apa yang sebenarnya baru?
Terdapat selalu memiliki algoritma yang baik untuk mempelajari bobot dalam jaringan dengan 1 lapisan tersembunyi

tetapi algoritma ini tidak pandai mempelajari bobot untuk jaringan dengan lapisan yang lebih tersembunyi



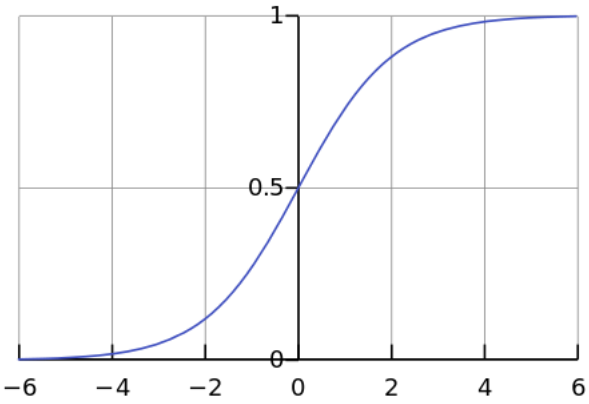
Apa yang terbaru: **algorithms for training many-later networks**



Jawaban yang lebih panjang

1. pengingat / penjelasan singkat tentang bagaimana bobot jaringan saraf dipelajari;
2. gagasan pembelajaran fitur tanpa pengawasan (***unsupervised feature learning***) (mengapa 'fitur menengah' penting untuk tugas klasifikasi yang sulit, dan bagaimana NN tampaknya mempelajarinya secara alami)
3. 'Terobosan' - trik sederhana untuk melatih jaringan saraf yang dalam

$$f(x) = \frac{1}{1 + e^{-x}}$$



-0.06

W1

-2.5

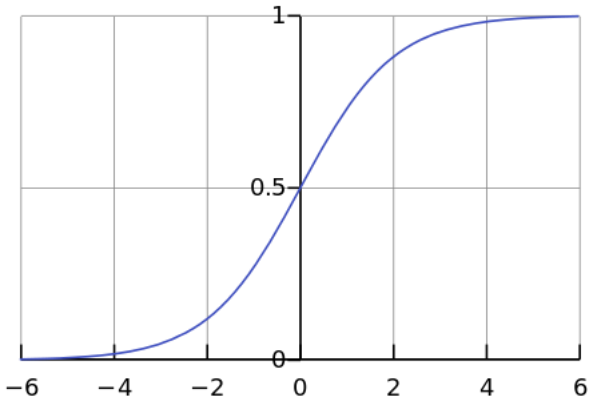
W2

W3

1.4

$f(x)$

$$f(x) = \frac{1}{1 + e^{-x}}$$

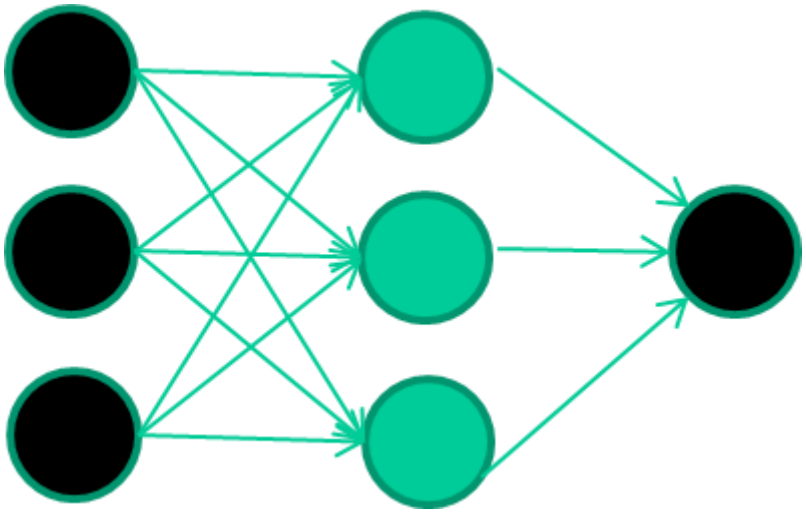


$x = -0.06 \times 2.7 + 2.5 \times 8.6 + 1.4 \times 0.002 = 21.34$

A dataset

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

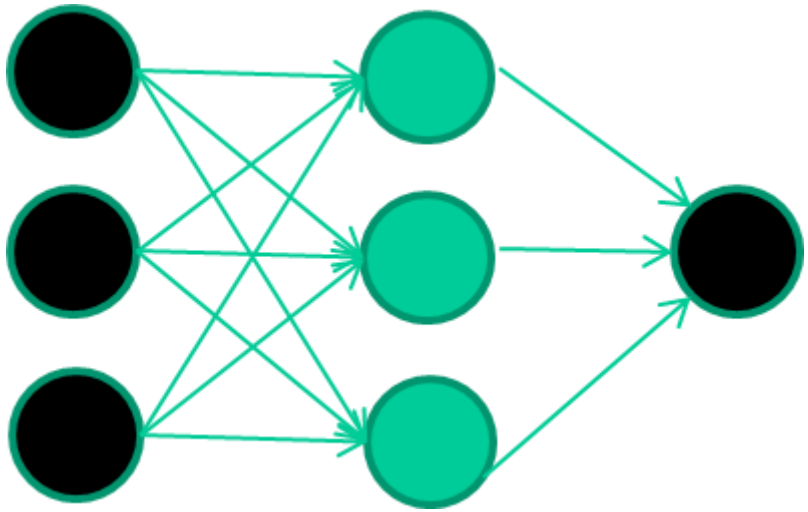
etc …



Training the neural network

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

etc …

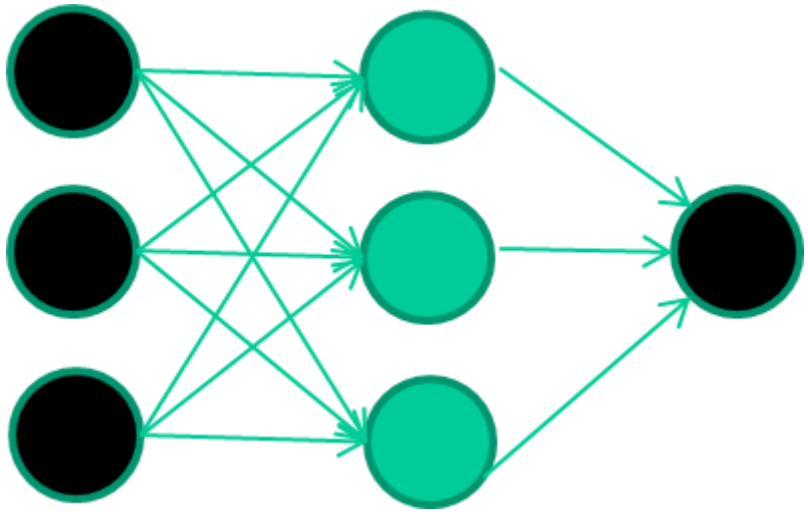


Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

etc …

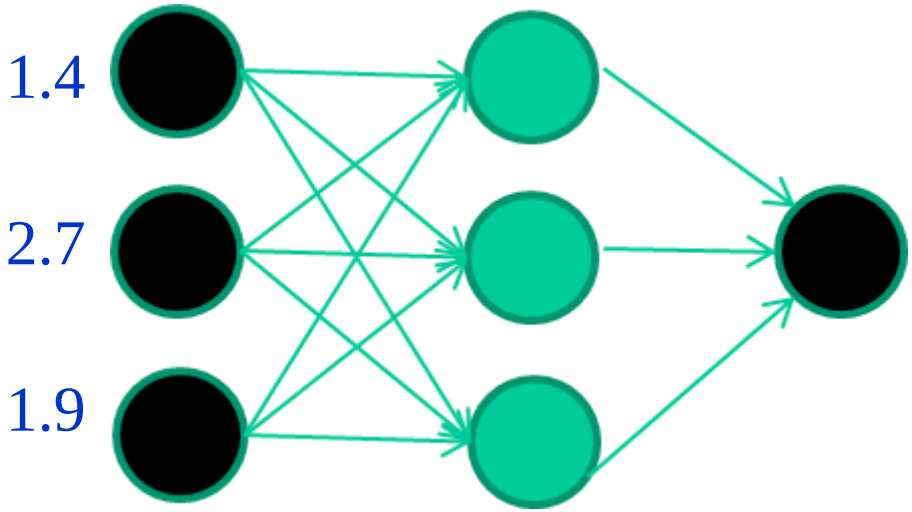
Initialise with random weights



Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

Present a training pattern

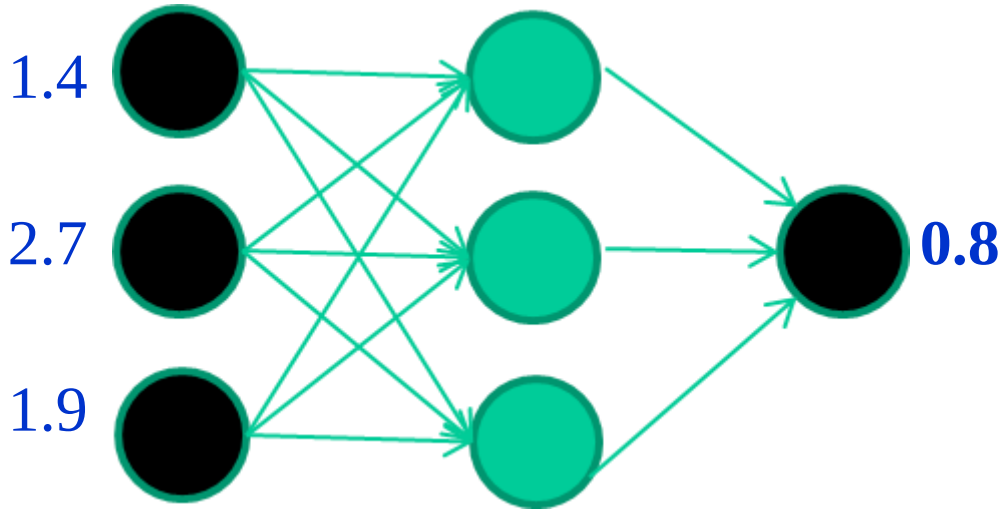


Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

etc …

Feed it through to get output

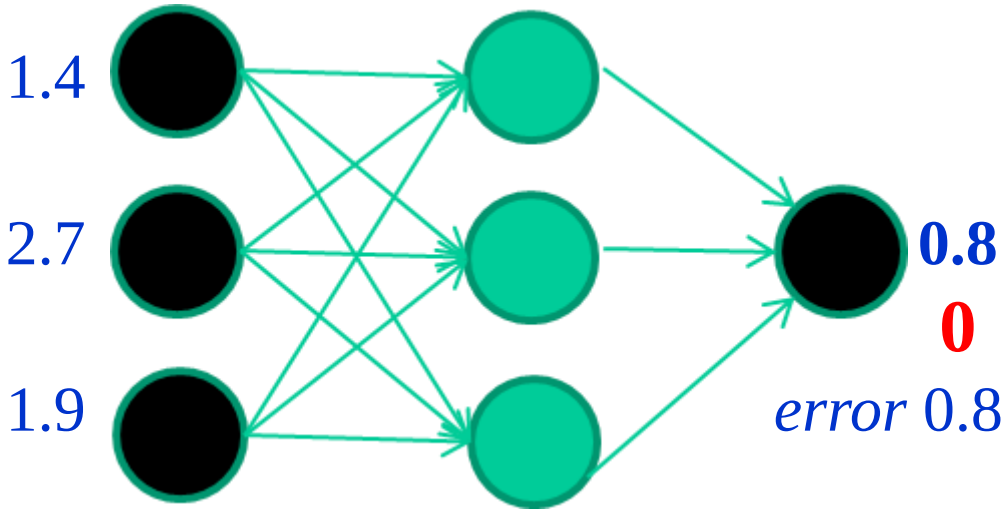


Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

etc …

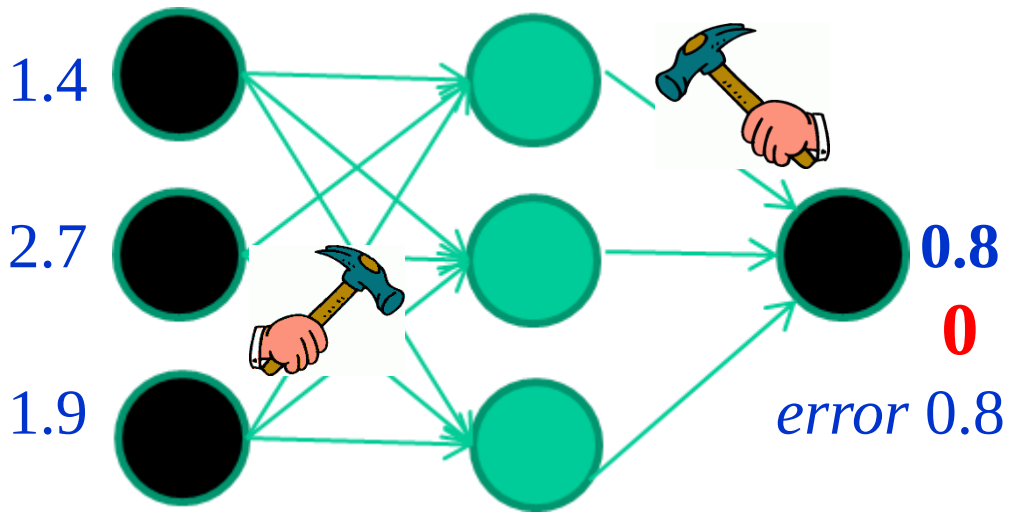
Compare with target output



Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

Adjust weights based on error

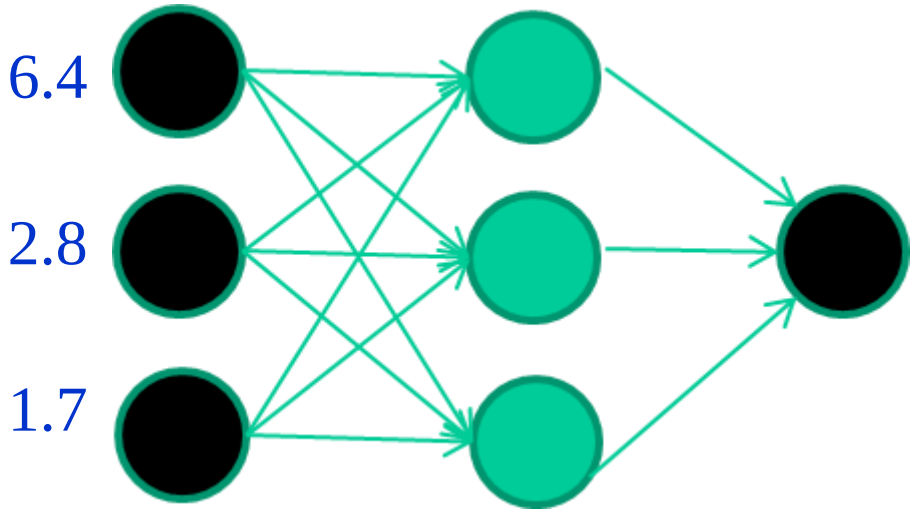


Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

etc …

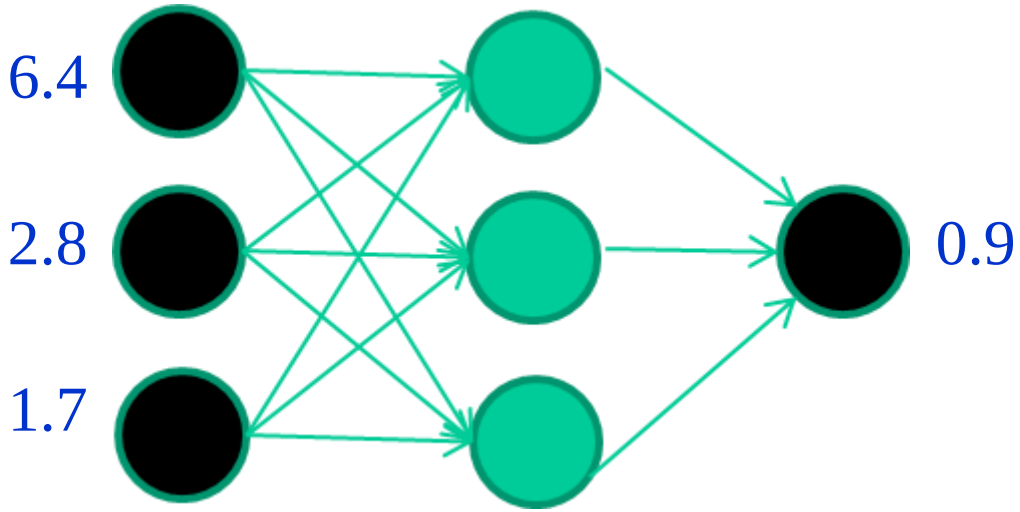
Present a training pattern



Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

Feed it through to get output

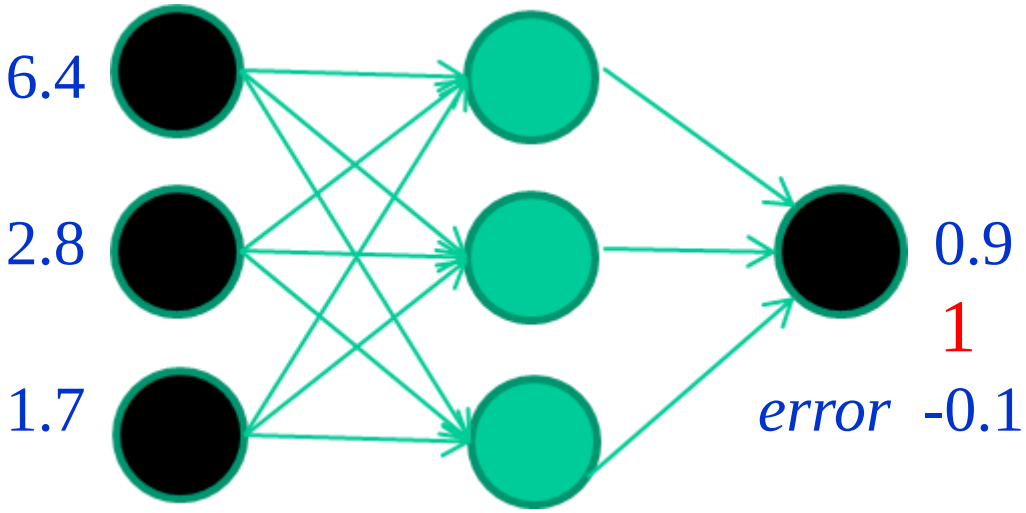


Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

etc …

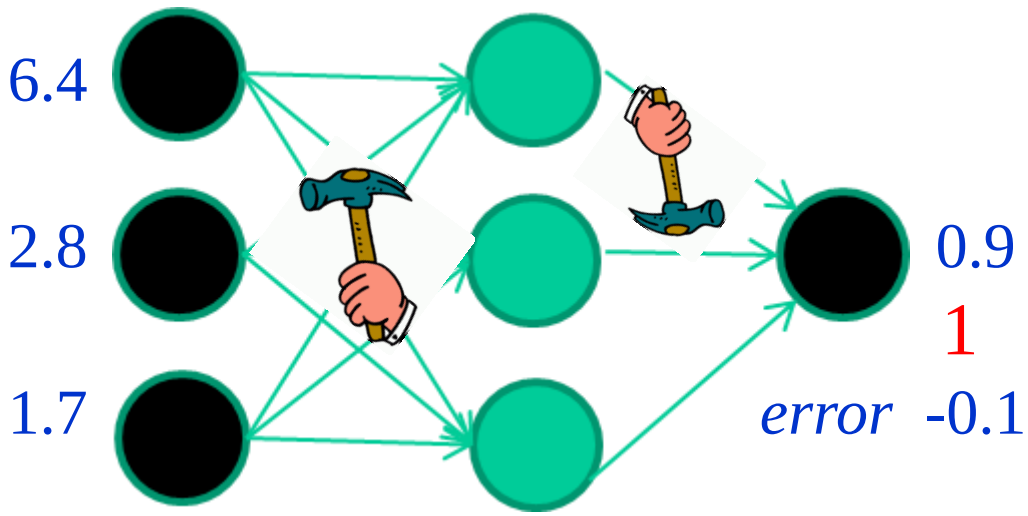
Compare with target output



Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			

Adjust weights based on error

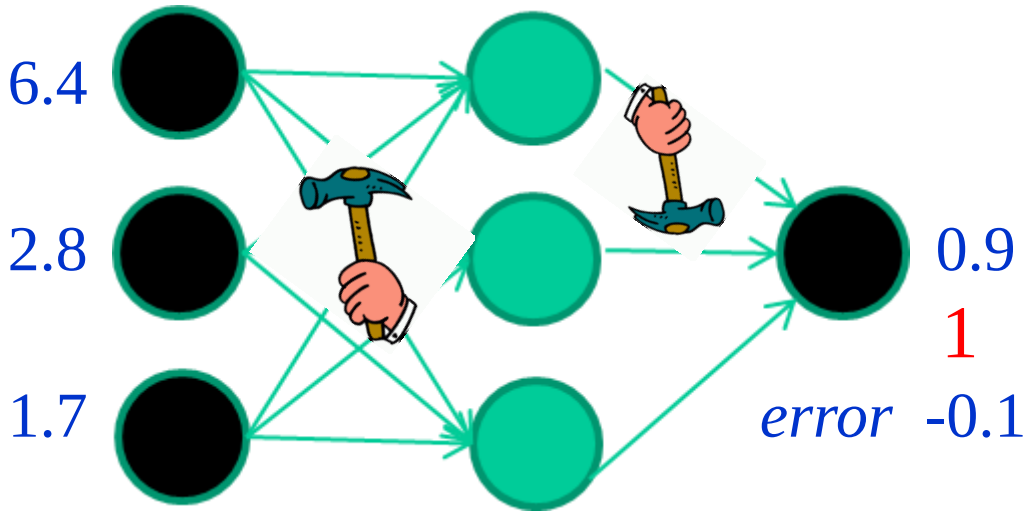


Training data

Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0

etc …

And so on ….

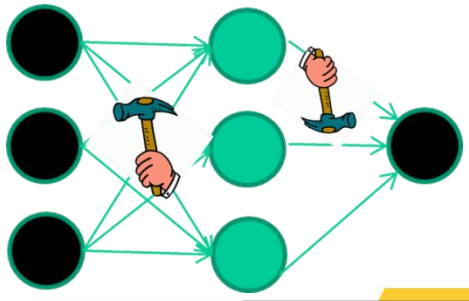


Ulangi ini ribuan, mungkin jutaan kali - setiap kali mengambil contoh pelatihan acak, dan membuat sedikit penyesuaian berat badan

Algoritma untuk penyesuaian berat dirancang untuk membuatnya perubahan yang akan mengurangi kesalahan

Inti yang akan dilakukan

- weight-learning algorithms (algoritma pembelajaran berat) untuk NNs are dumb
- mereka bekerja dengan membuat ribuan penyesuaian kecil, masing-masing membuat jaringan melakukan lebih baik pada pola terbaru, tetapi mungkin sedikit lebih buruk pada banyak lainnya
- tetapi, karena keberuntungan, akhirnya ini cenderung cukup baik pelajari pengklasifikasi yang efektif untuk banyak aplikasi nyata



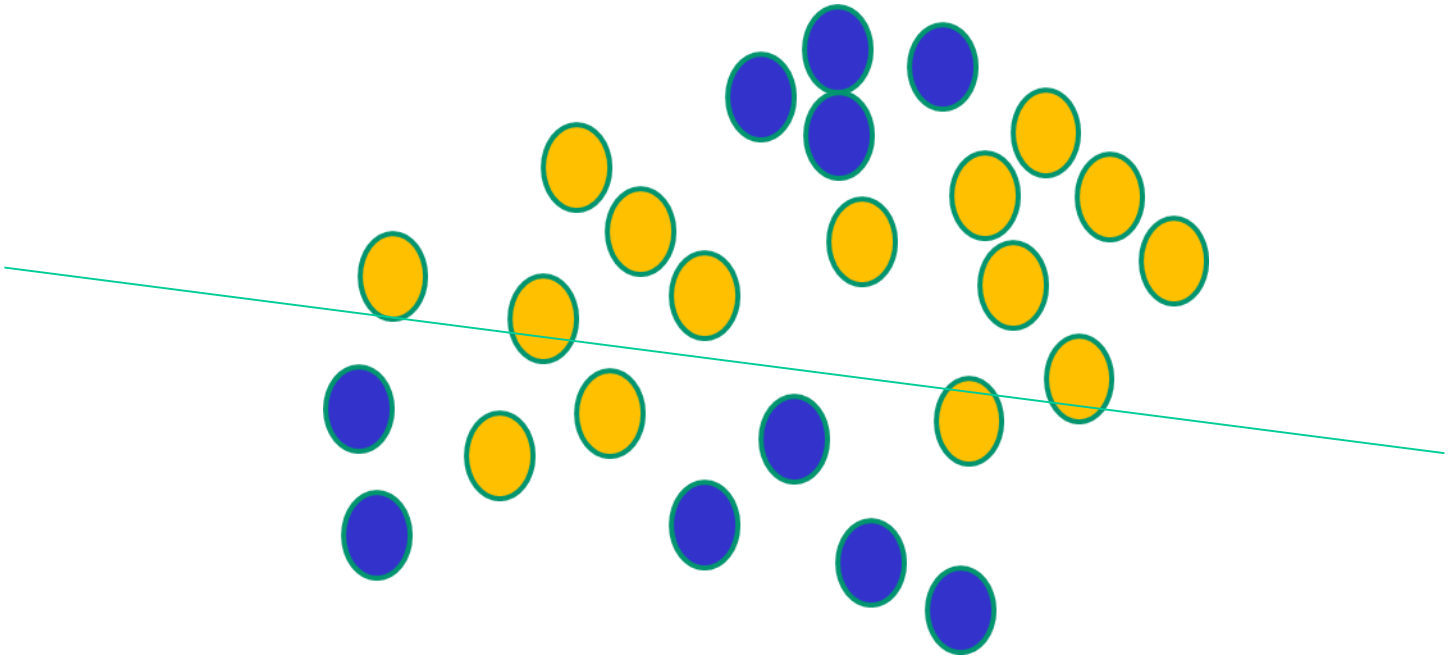
Beberapa inti lainnya

Detail dari standar algoritma pembelajaran bobot NN - nanti

Jika f (x) non-linear, jaringan dengan 1 lapisan tersembunyi, secara teori, dapat mempelajari dengan sempurna masalah klasifikasi. Ada set bobot yang dapat menghasilkan target dari input. Masalahnya adalah menemukan nya.

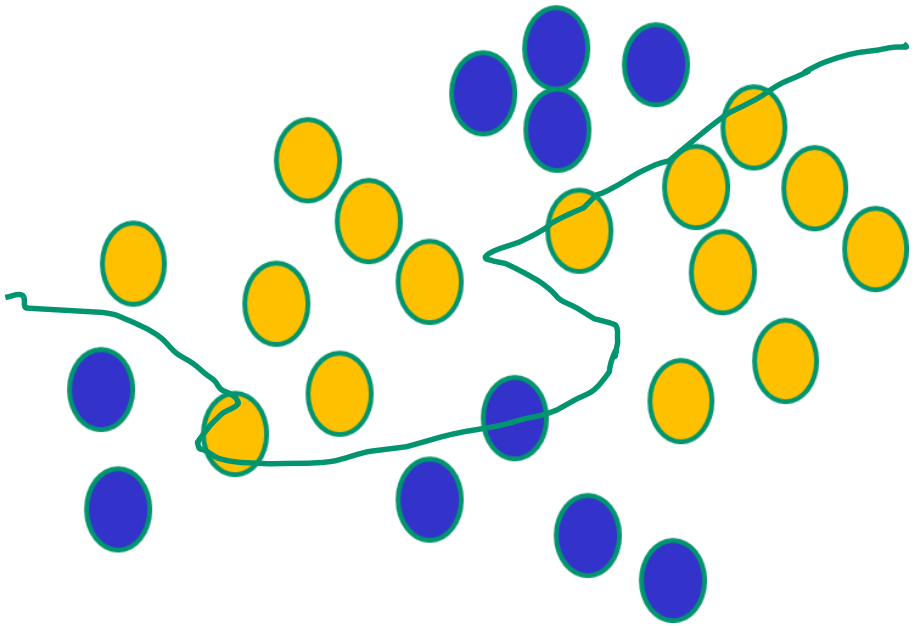
Beberapa poin lain nya

Jika f (x) linier, NN hanya dapat menggambar batas keputusan langsung (bahkan jika ada banyak lapisan unit)



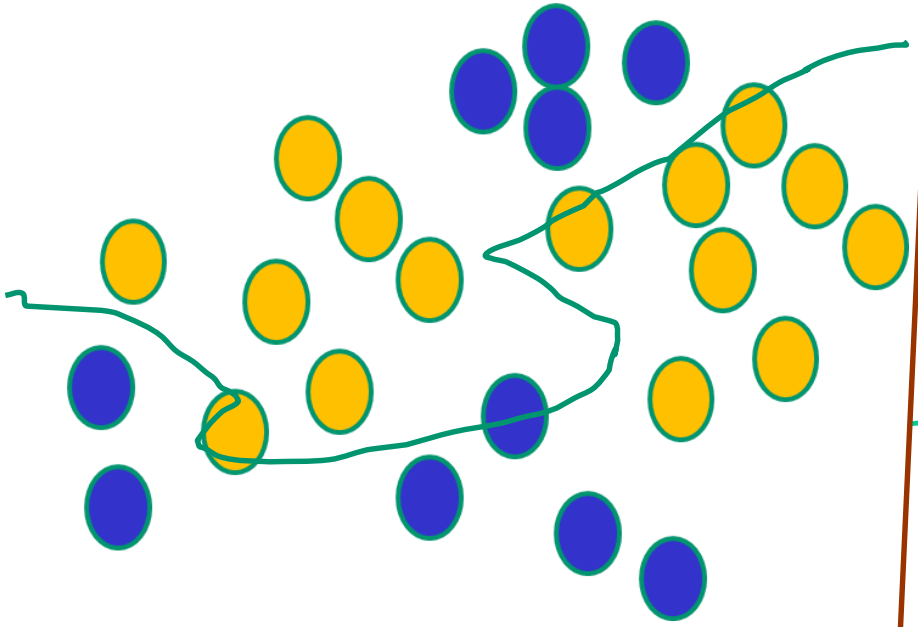
Beberapa poin lain nya

NN menggunakan nonlinear f (x) sehingga mereka dapat menggambar batas yang kompleks, tetapi menjaga data tidak berubah

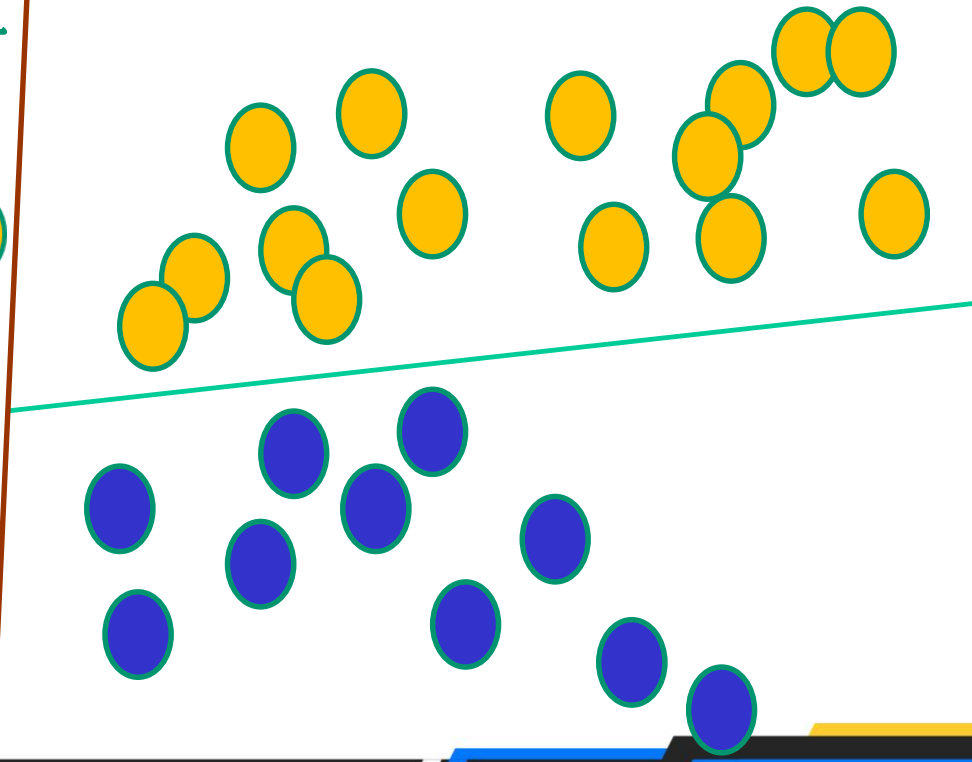


Beberapa poin lain nya

NN menggunakan nonlinear f (x) sehingga mereka dapat menggambar batas yang kompleks, tetapi menjaga data tidak berubah



SVM hanya menggambar garis lurus tetapi mereka mengubah data terlebih dahulu dengan cara yang membuat itu OK



Feature detectors

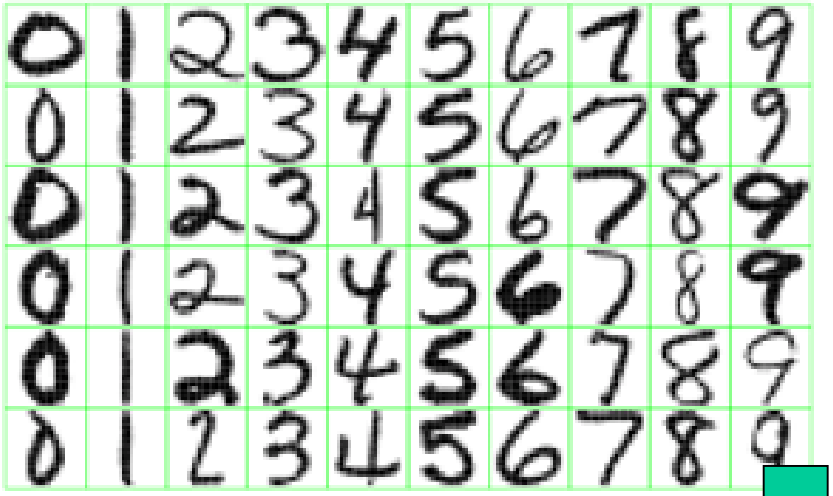
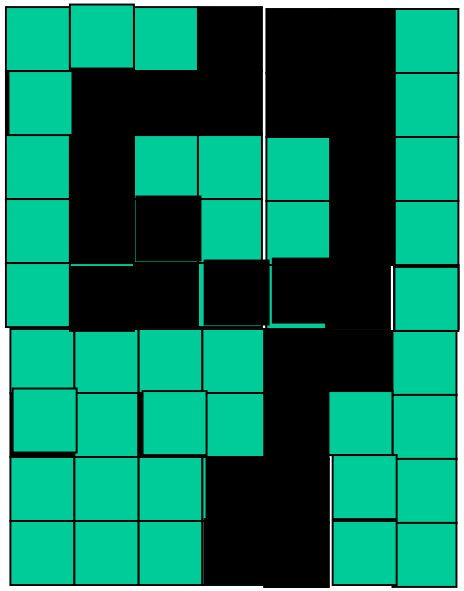
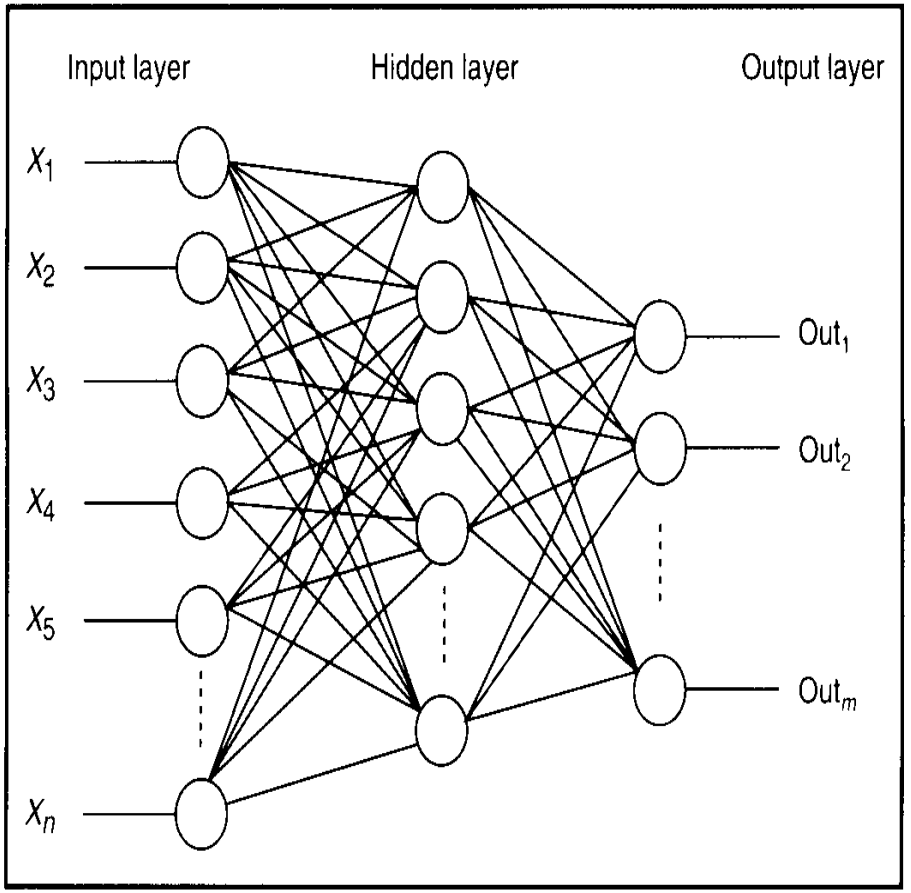


Figure 1.2: *Examples of handwritten digits from U.S. postal envelopes.*





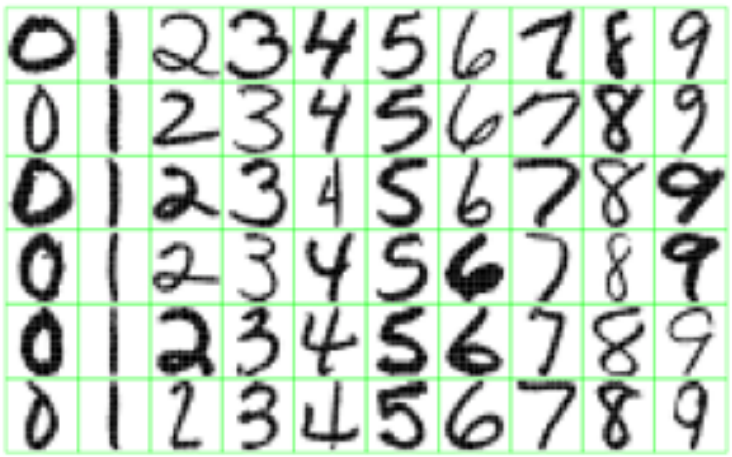
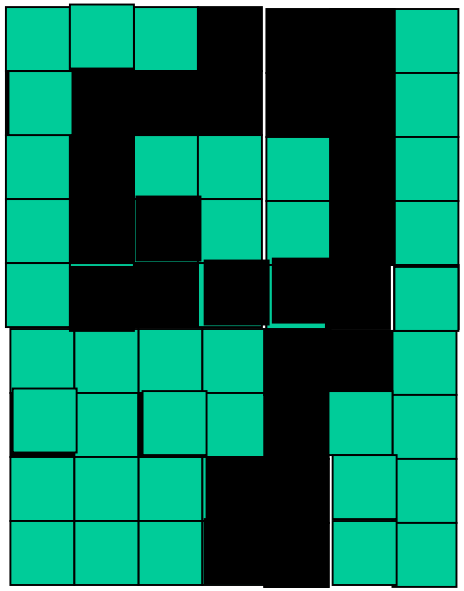
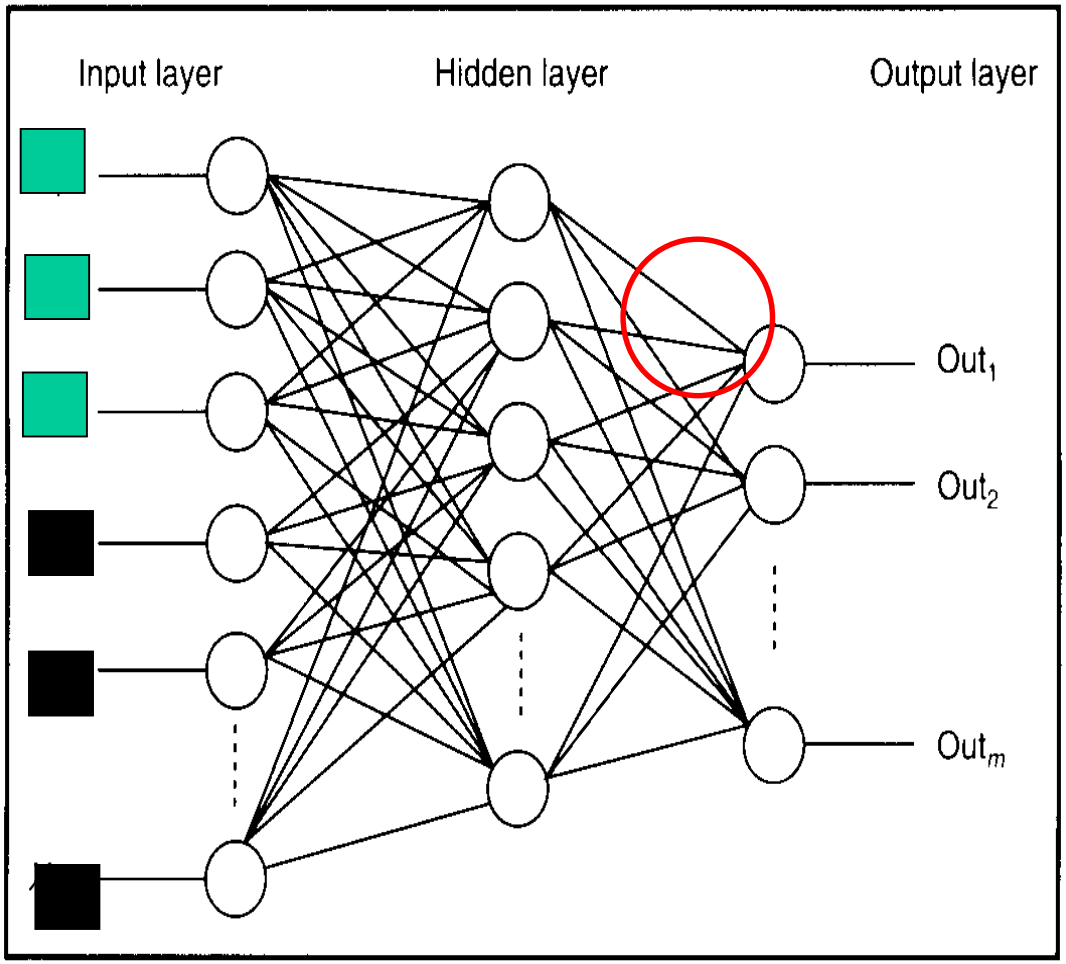


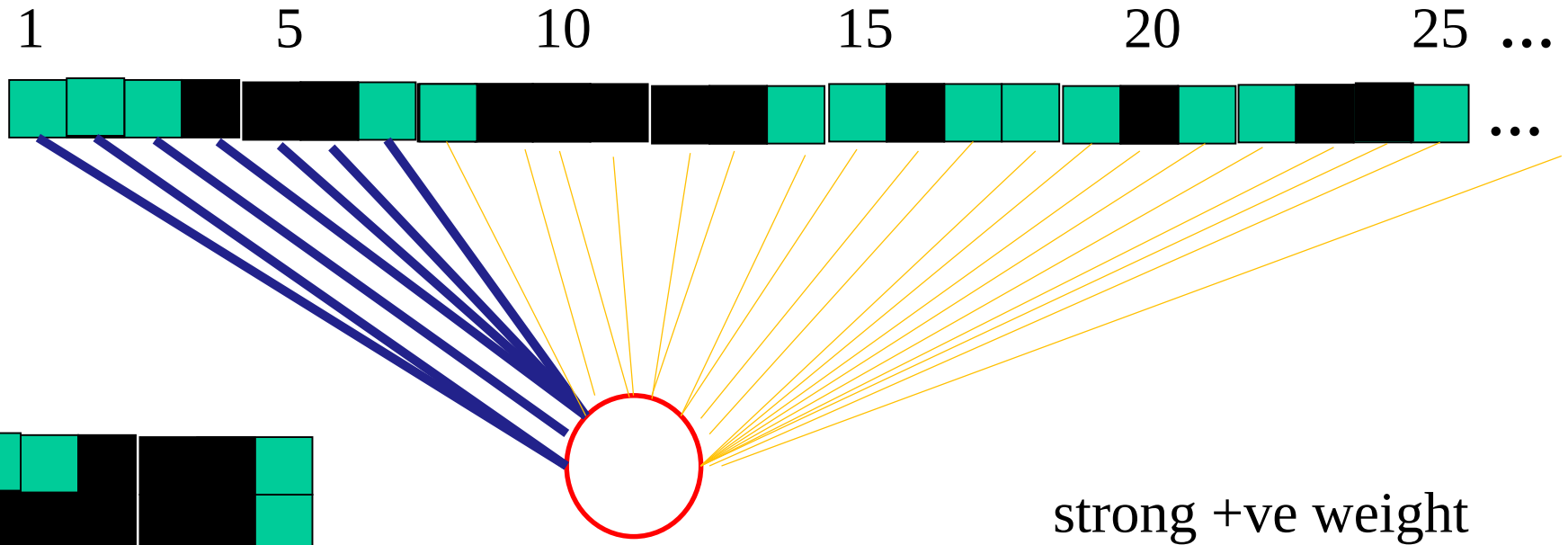
Figure 1.2: *Examples of handwritten digits from U.S. postal envelopes.*

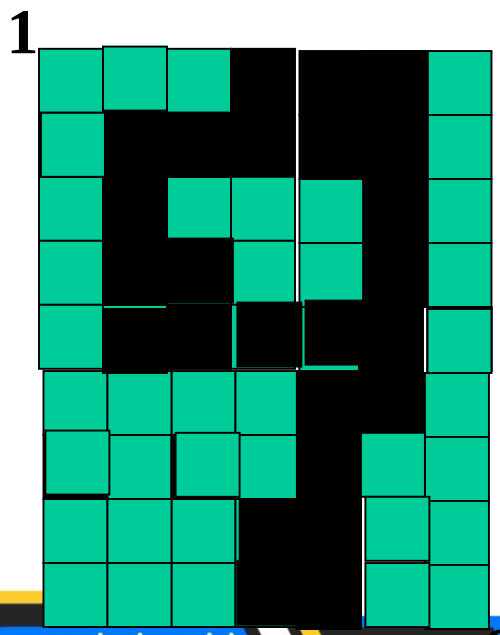


what is this unit doing?



Unit layer tersembunyi menjadi pendeteksi fitur yang diatur sendi

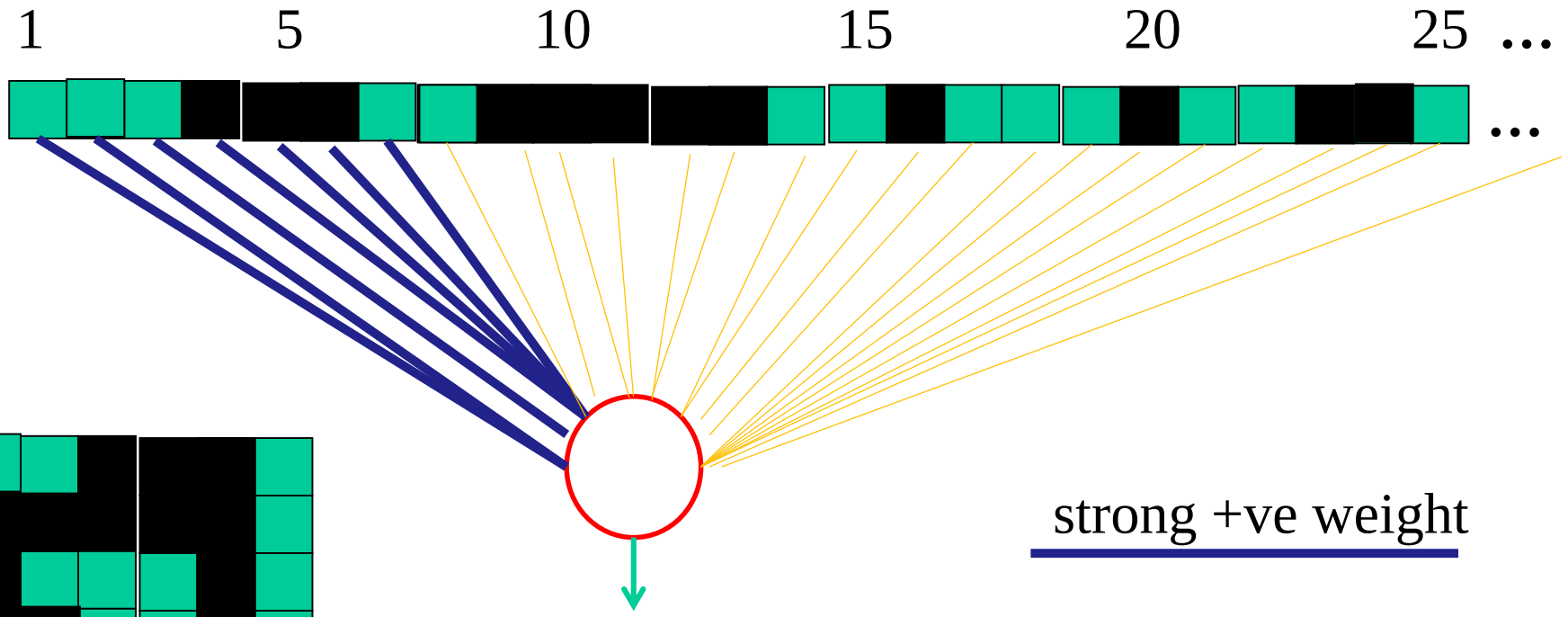


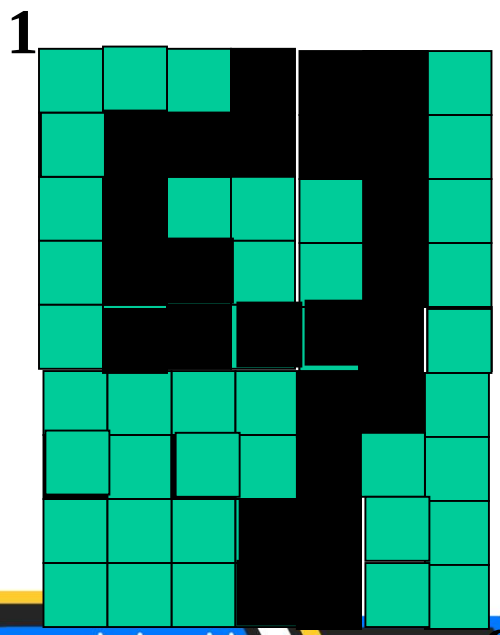


strong +ve weight

low/zero weight

Apa yang dideteksi unit ini?

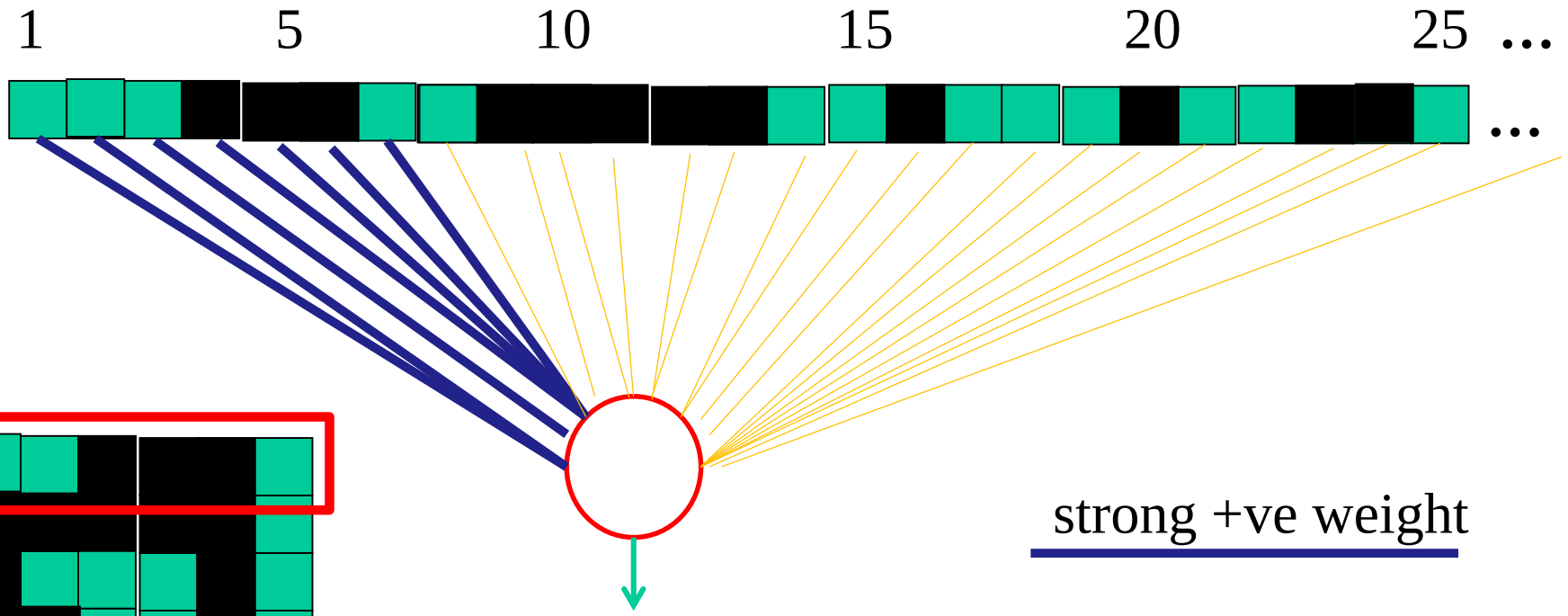


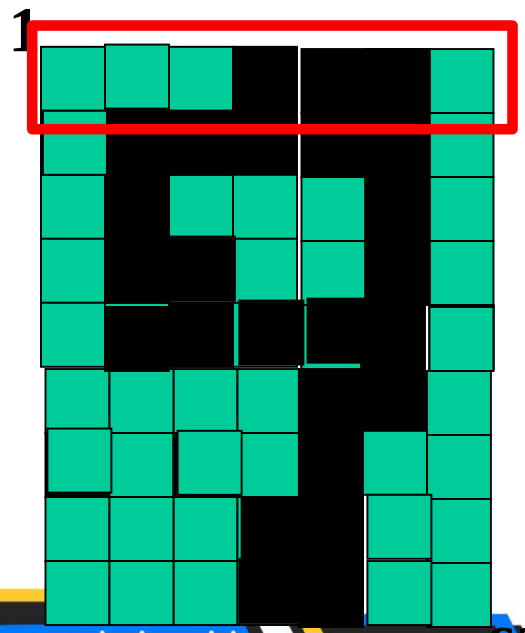


strong +ve weight

low/zero weight

Apa yang dideteksi unit ini?



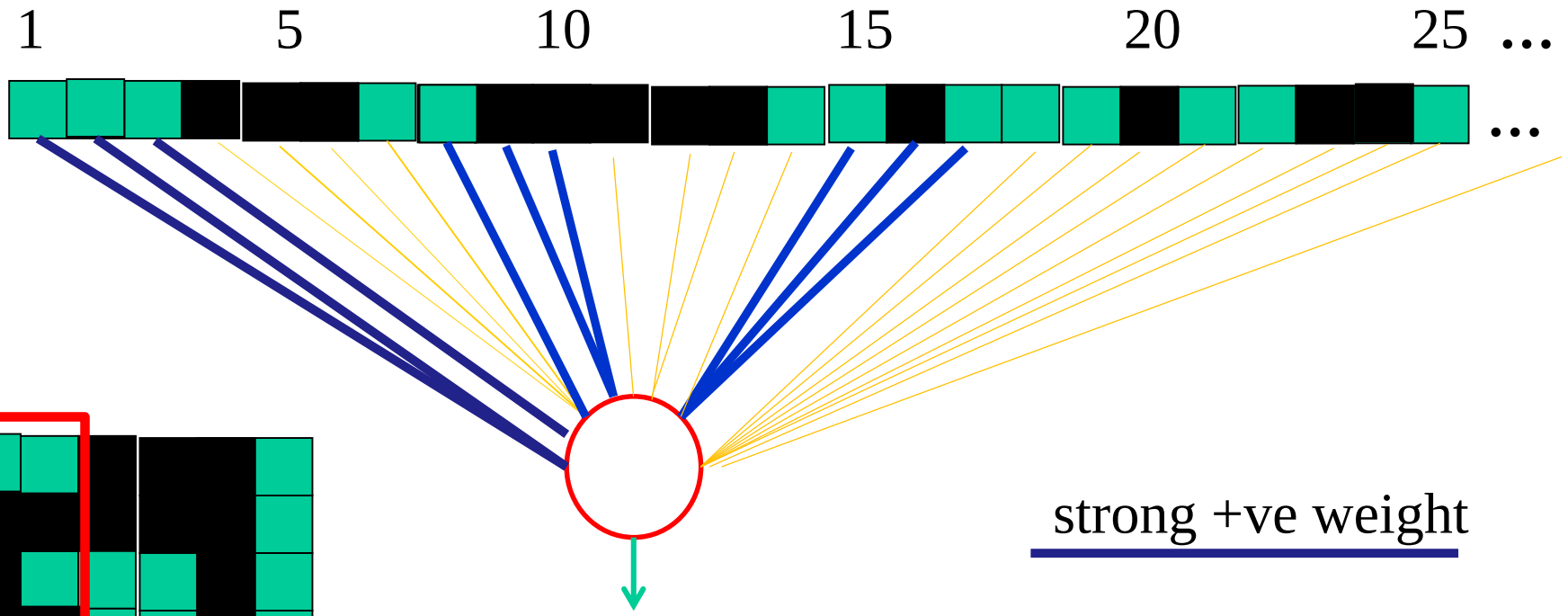


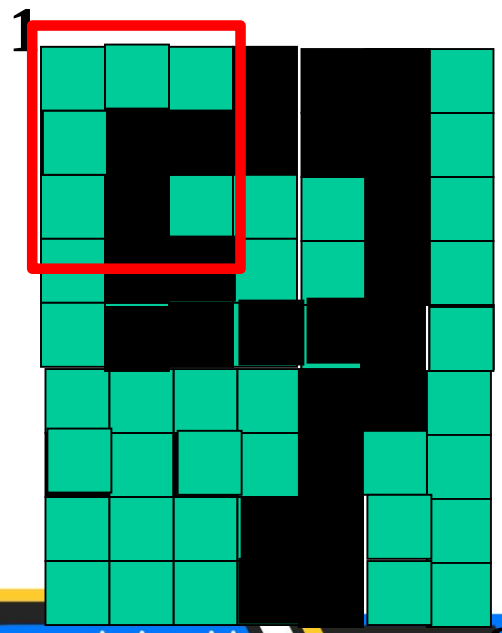
strong +ve weight

low/zero weight

it will send strong signal for a horizontal line in the top row, ignoring everywhere else

Apa yang dideteksi unit ini?

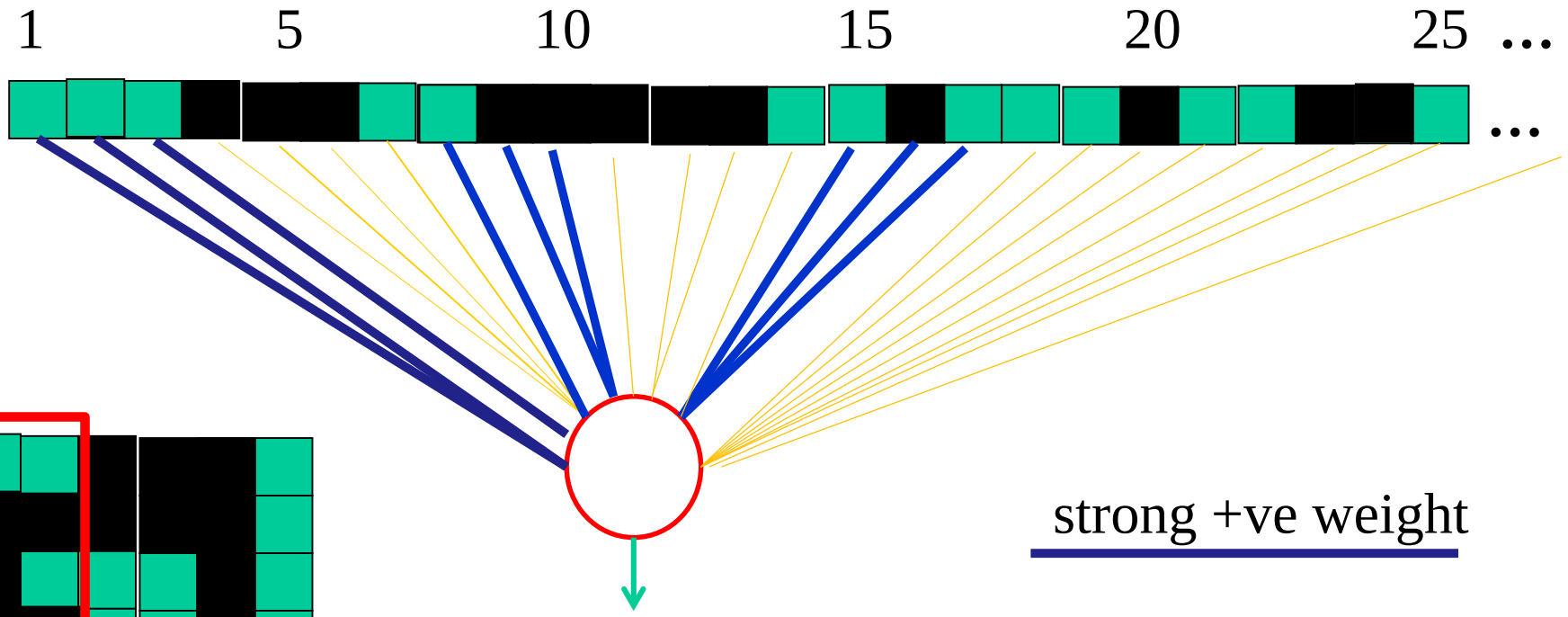


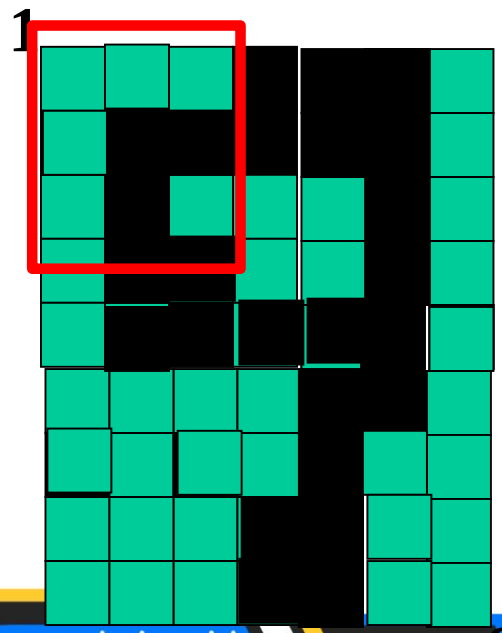


strong +ve weight

low/zero weight

Apa yang dideteksi unit ini?





strong +ve weight

low/zero weight

Strong signal for a dark area in the top left corner

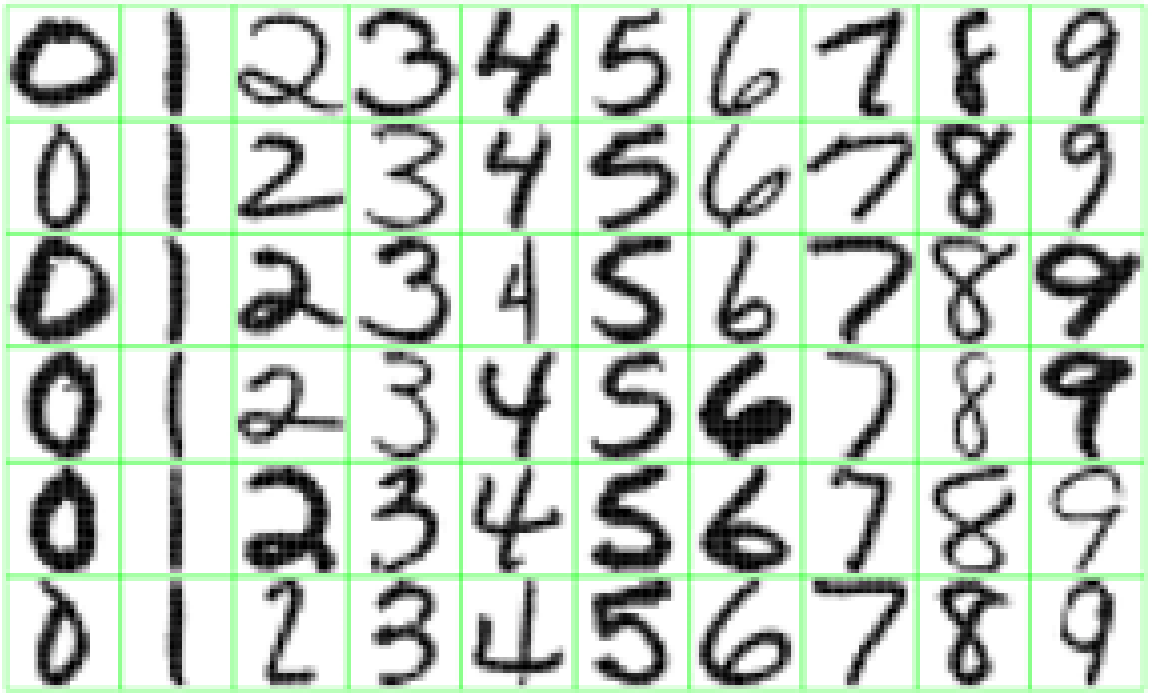
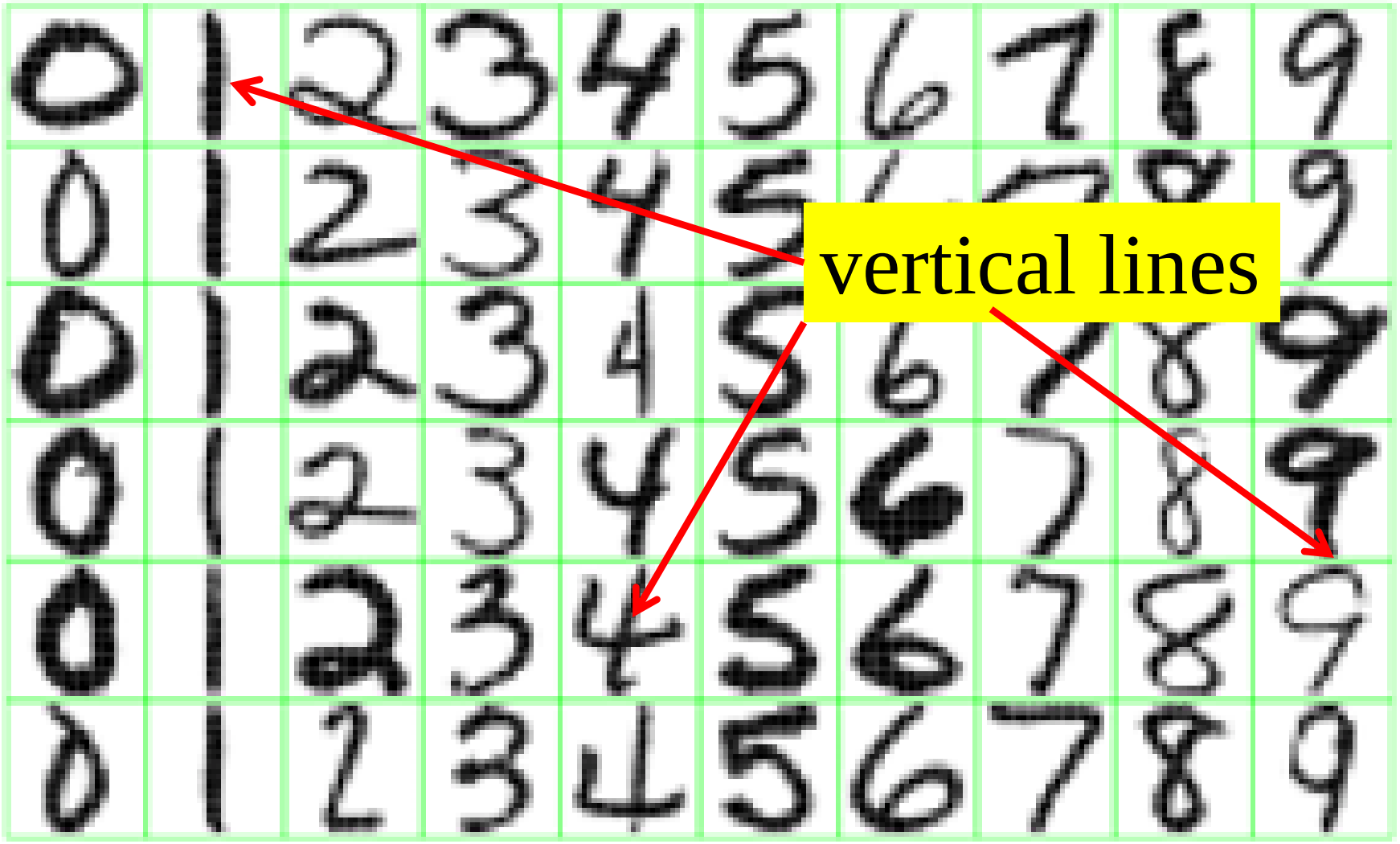


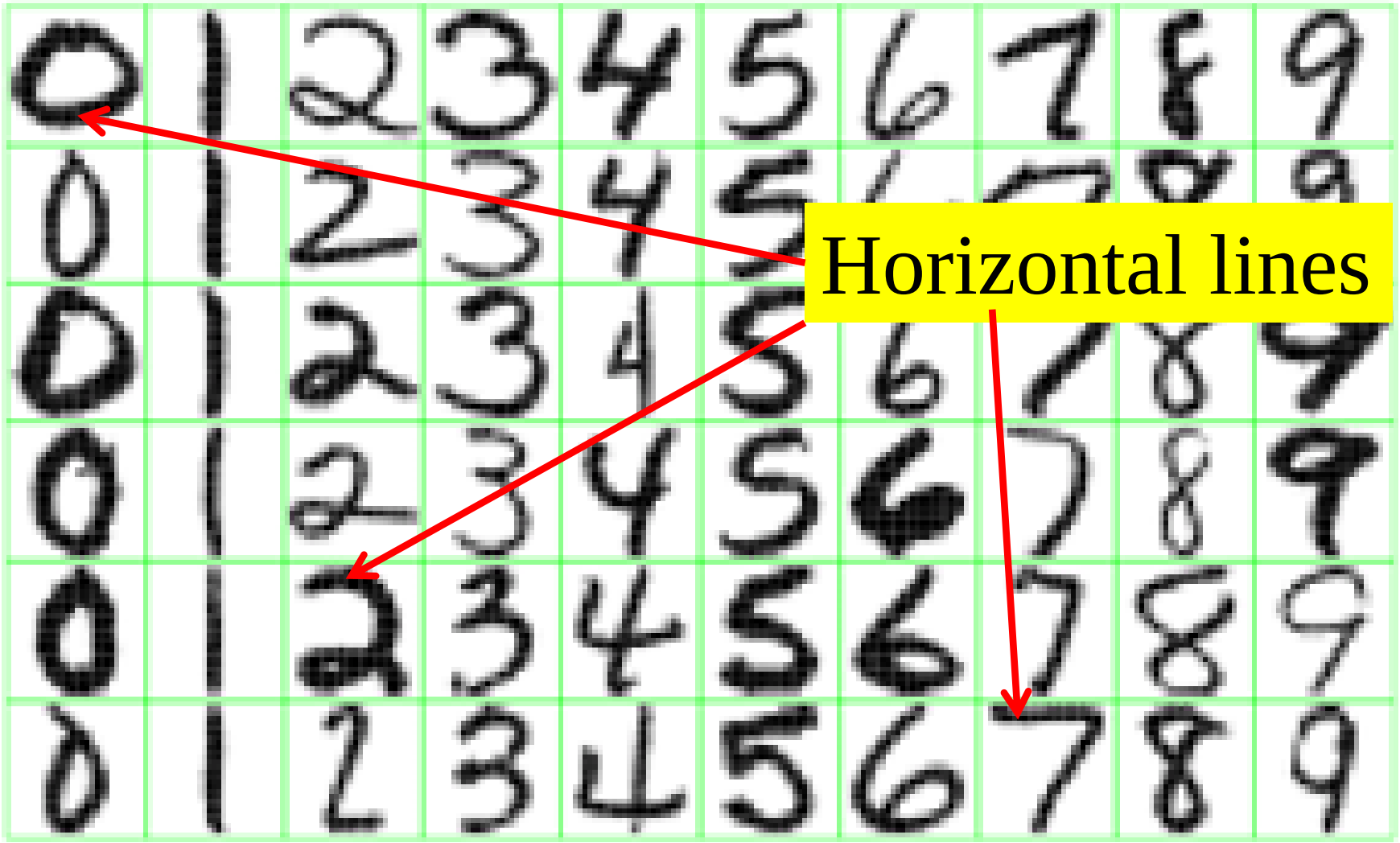
Figure 1.2: *Examples of handwritten digits from U.S. postal envelopes.*

What features might you expect a good NN to learn, when trained with data like this?



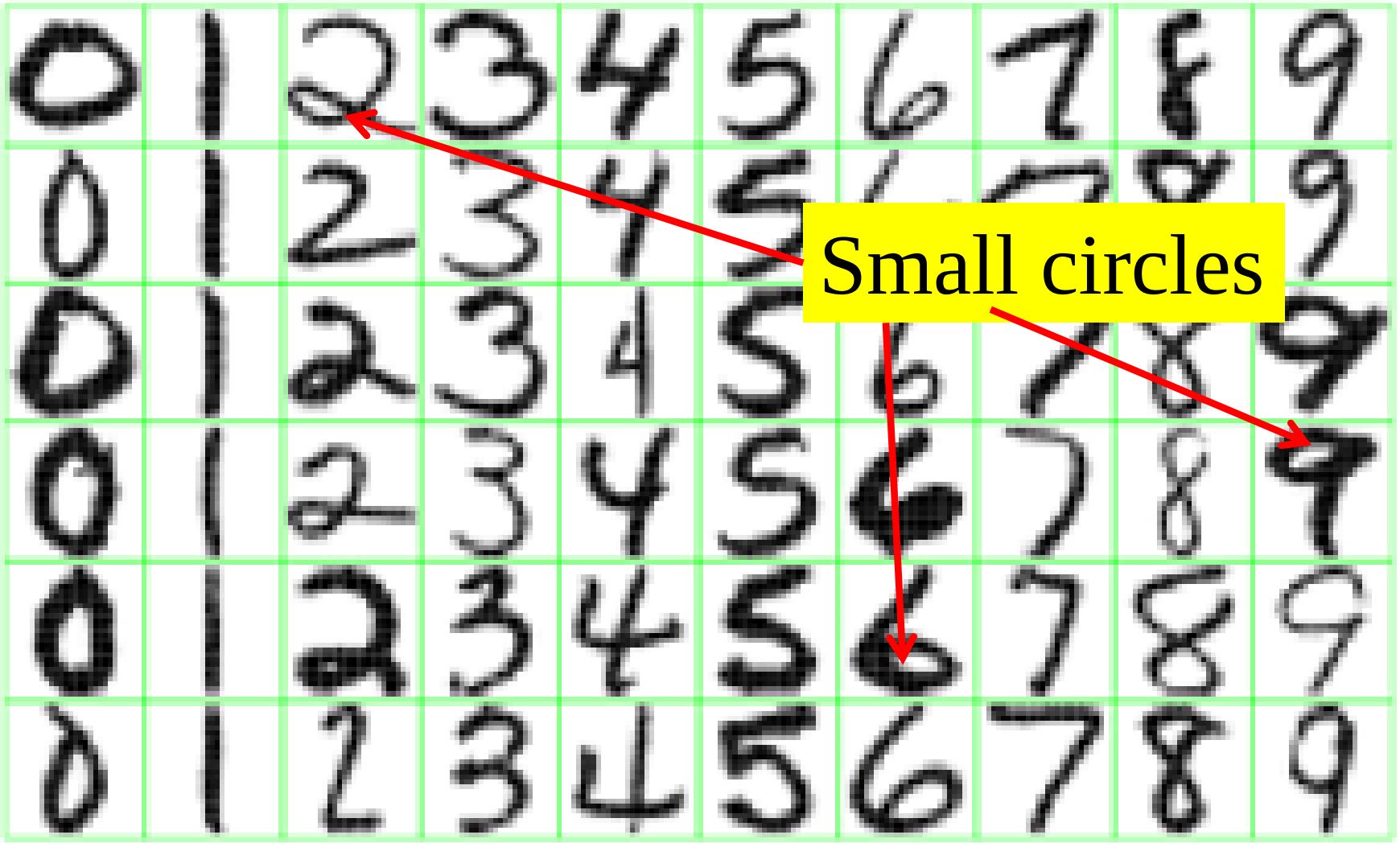
1

Figure 1.2: *Examples of handwritten digits from U.S. postal envelopes.*



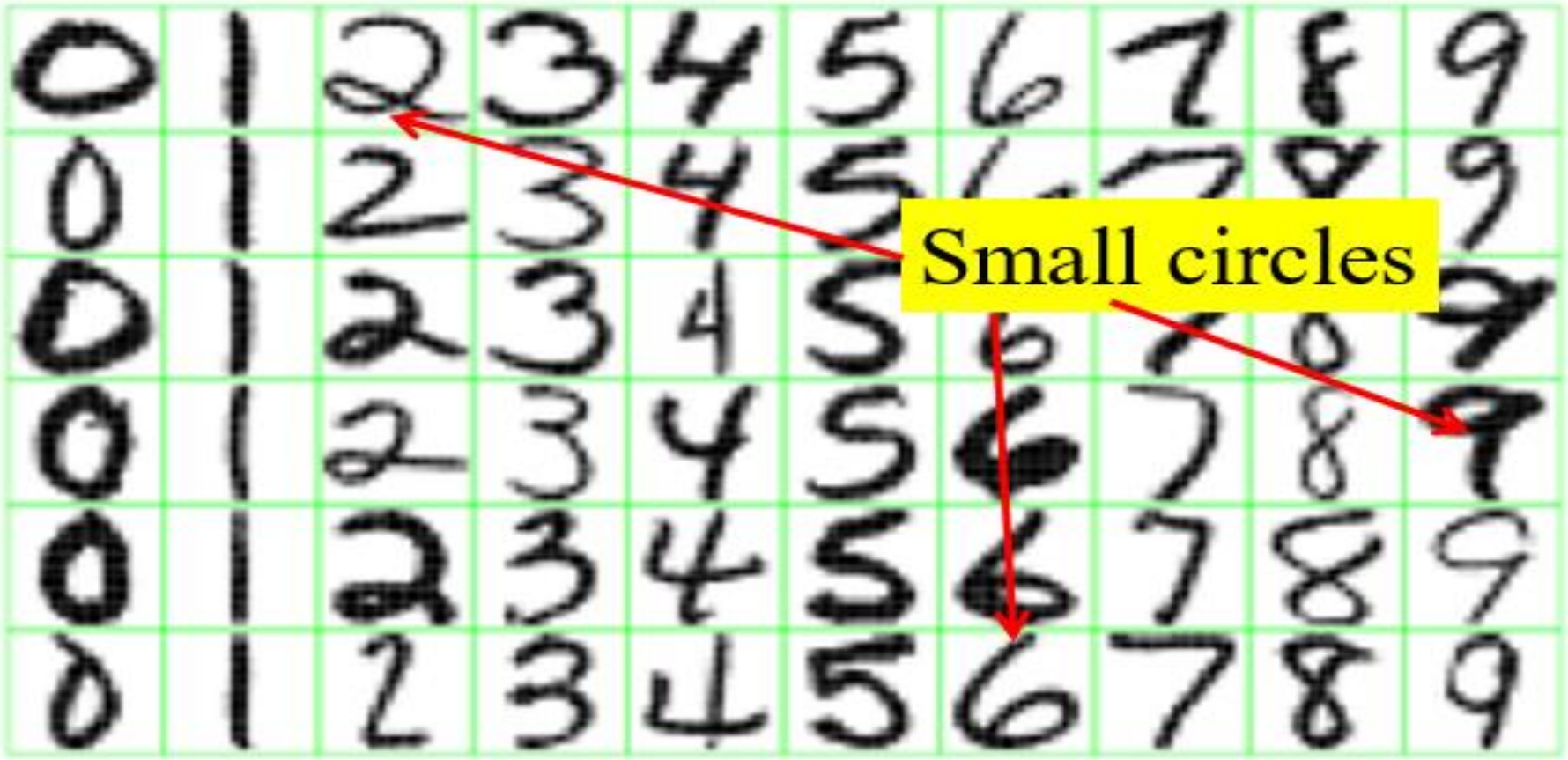
1

Figure 1.2: *Examples of handwritten digits from U.S. postal envelopes.*



1

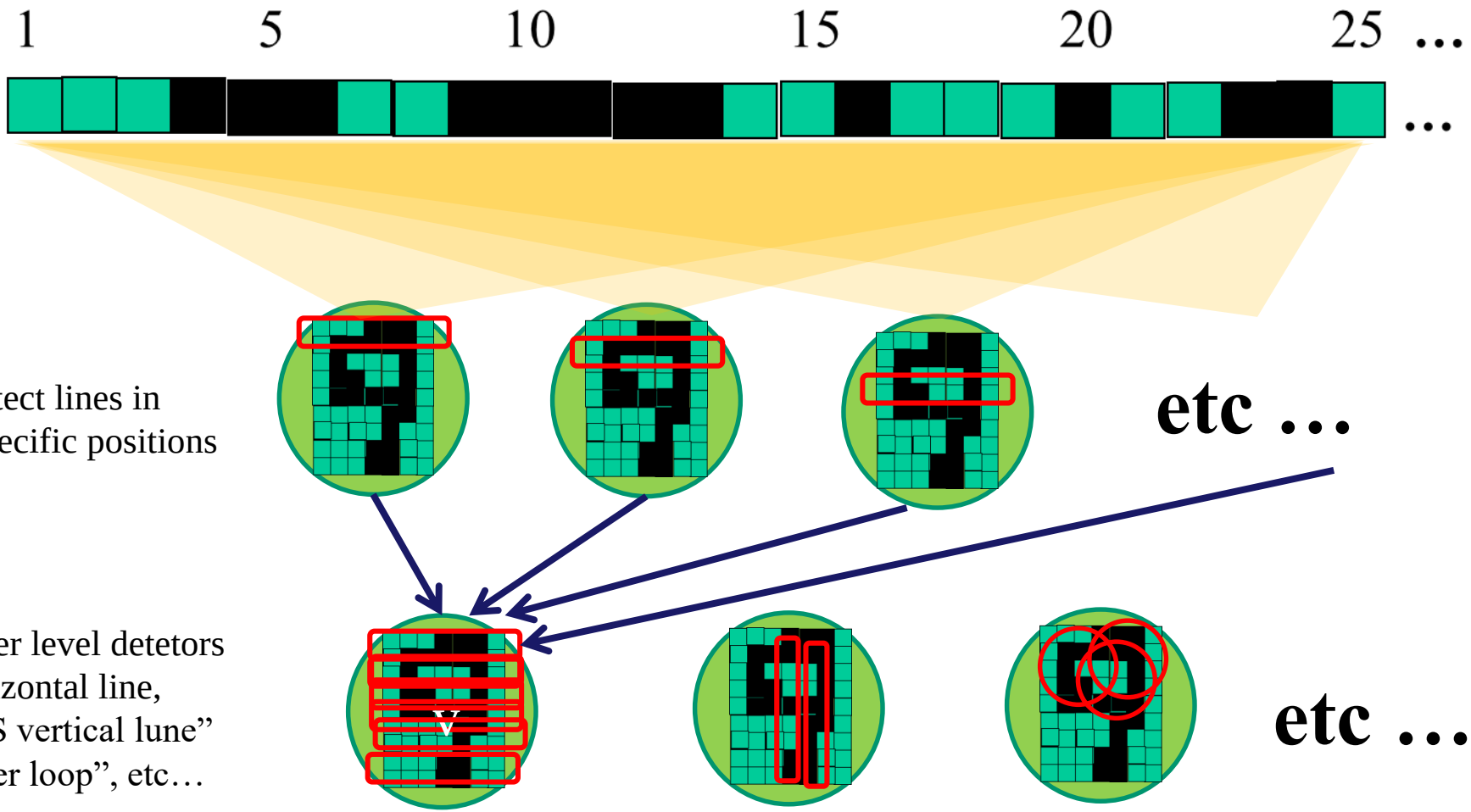
Figure 1.2: *Examples of handwritten digits from U.S. postal envelopes.*



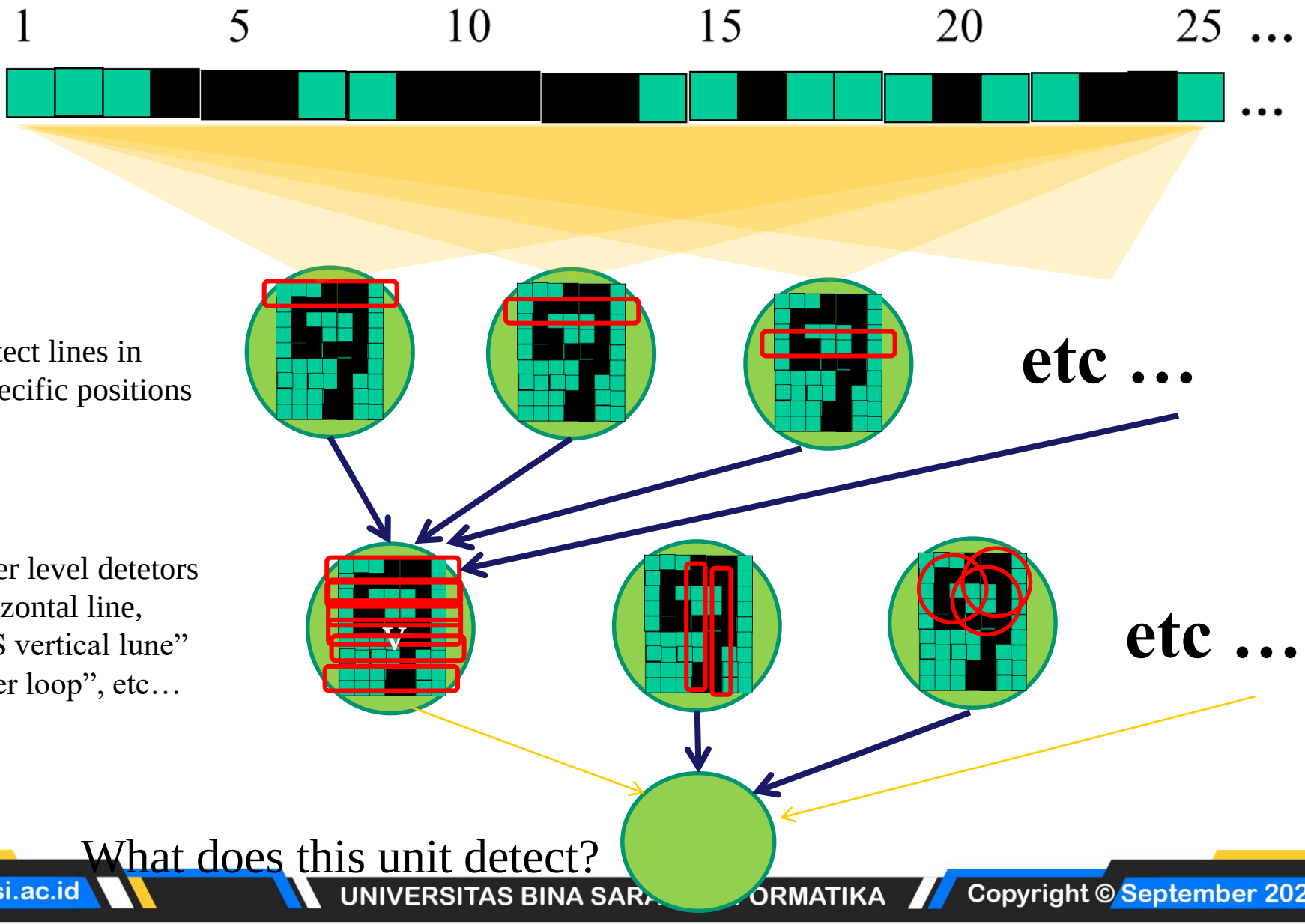
1

Tapi bagaimana dengan invarian posisi ???
detektor unit contoh diatas terikat bagian tertentu dari gambar

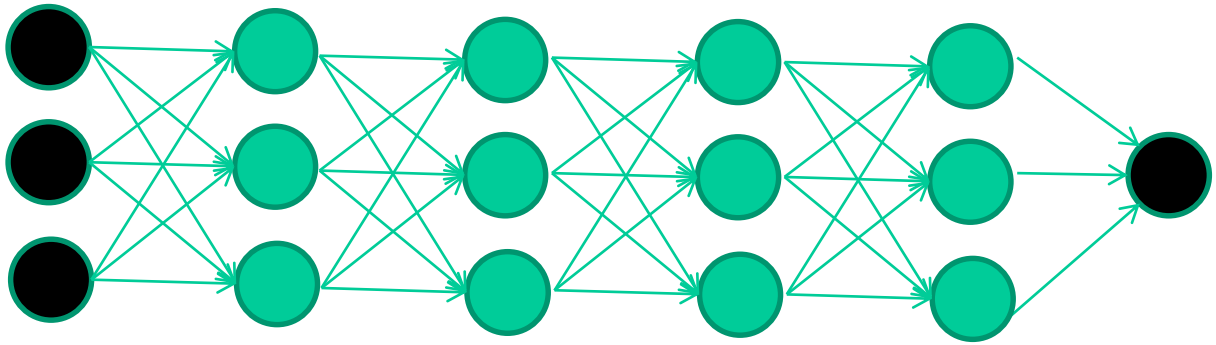
successive layers can learn higher-level features …



successive layers can learn higher-level features …

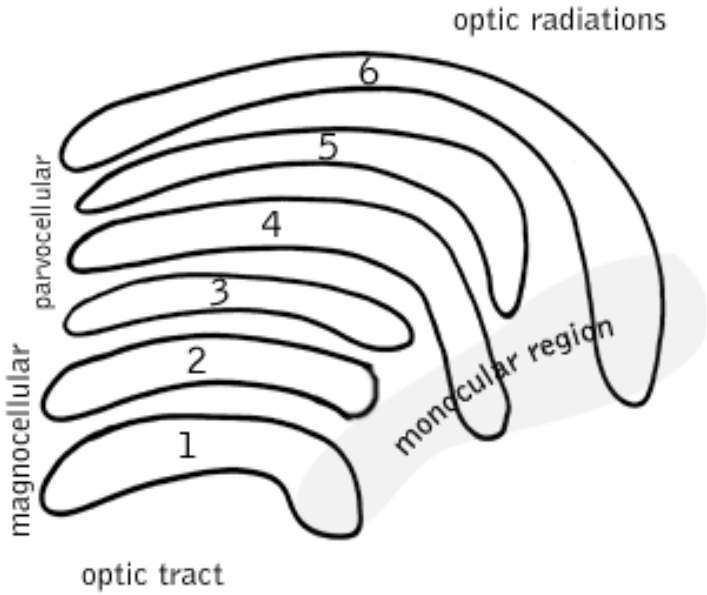


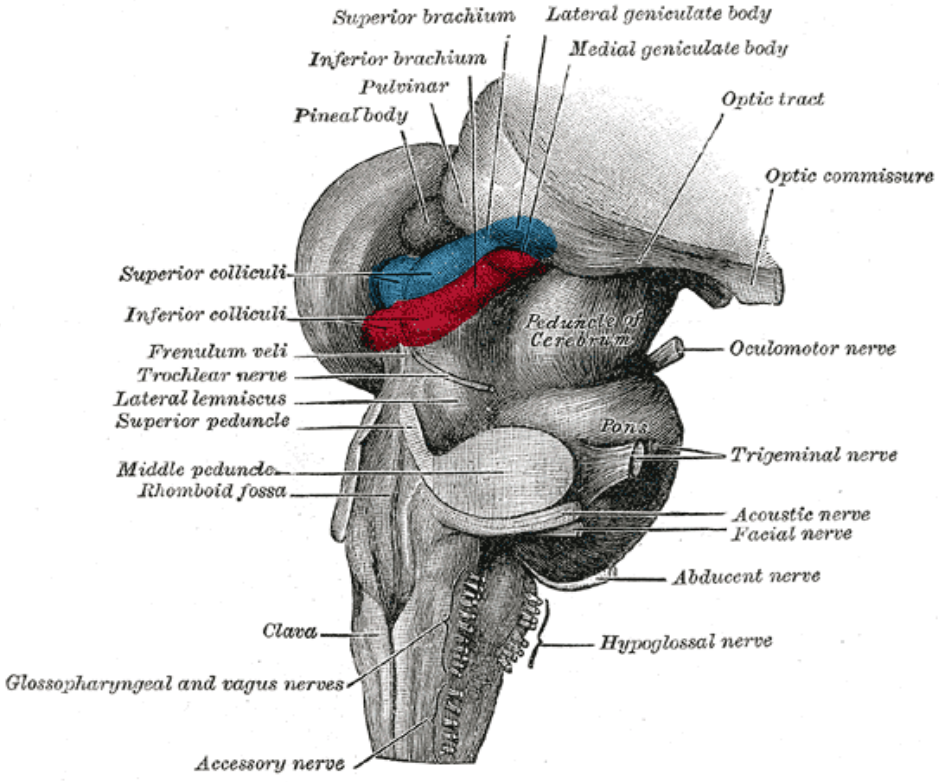
So: *multiple layers make sense*



So: *multiple layers make sense*

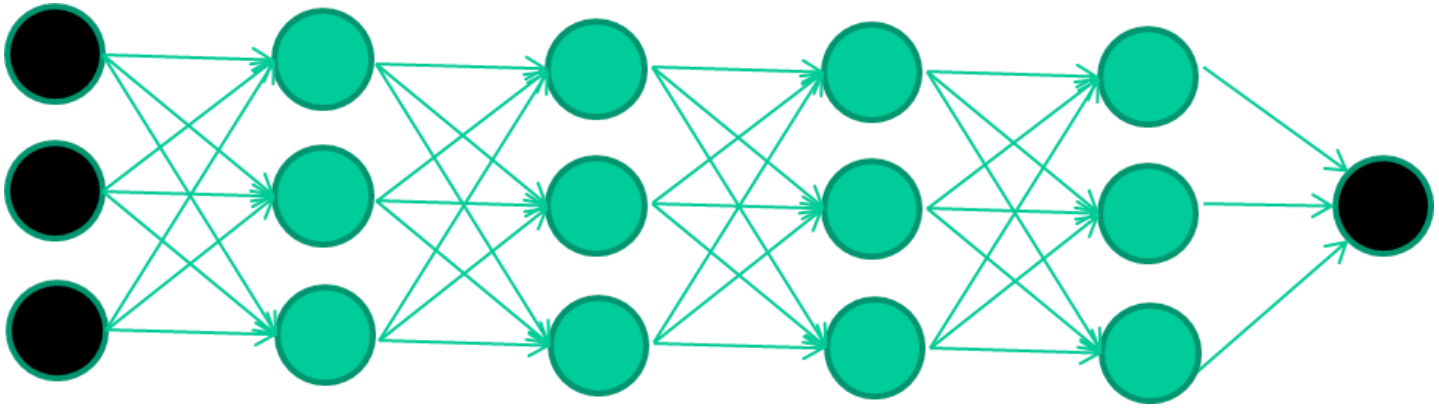
Your brain works that way



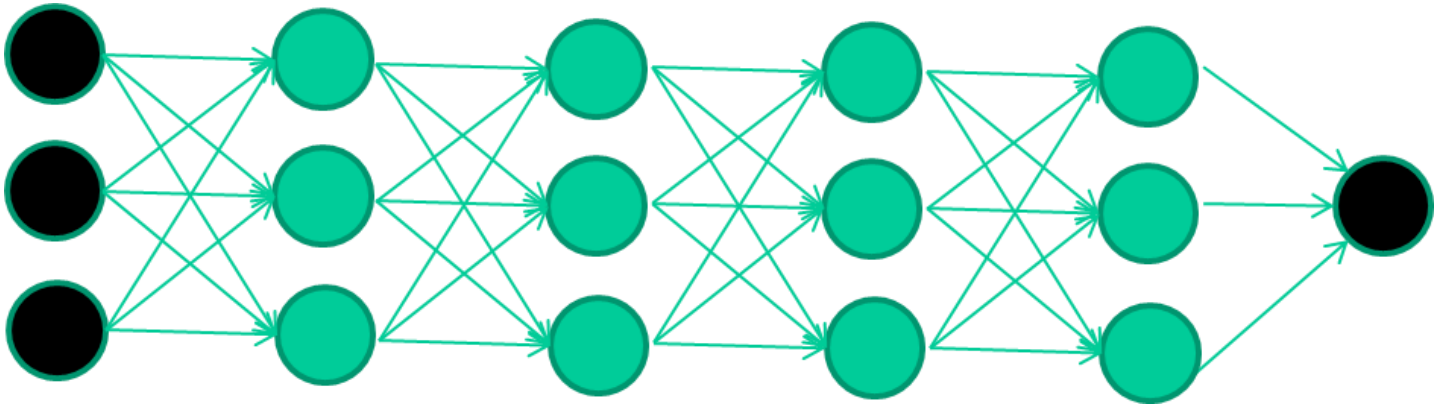


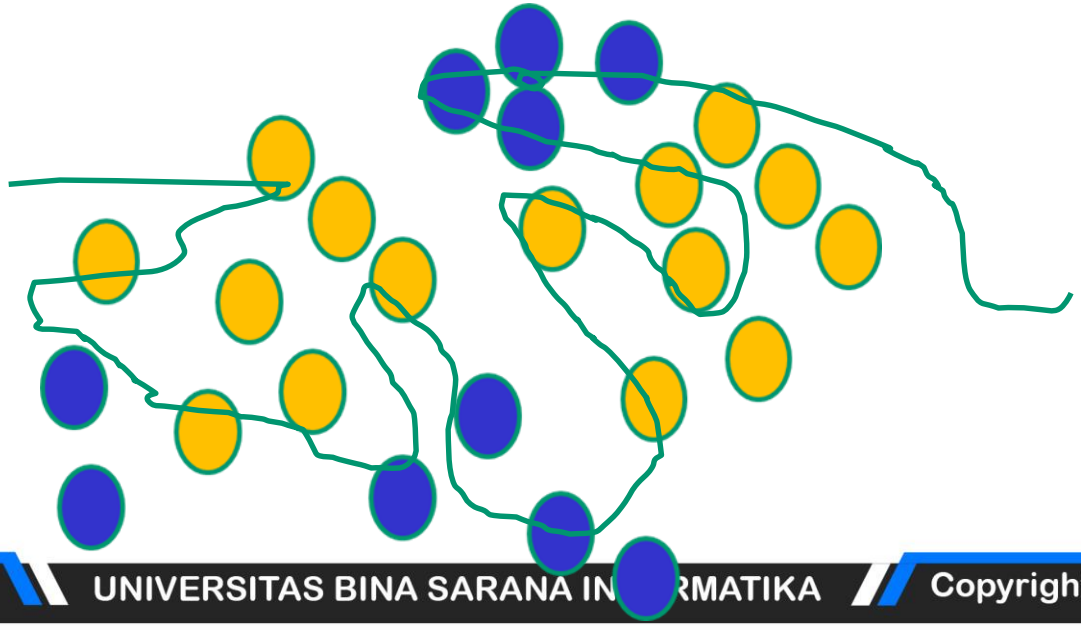
So: *multiple layers make sense*

Banyak-lapisan Arsitektur Neural Network (jaringan saraf) harus mampu mempelajari fitur-fitur mendasar yang sebenarnya dan 'logika fitur', dan karenanya menggeneralisasi dengan sangat baik



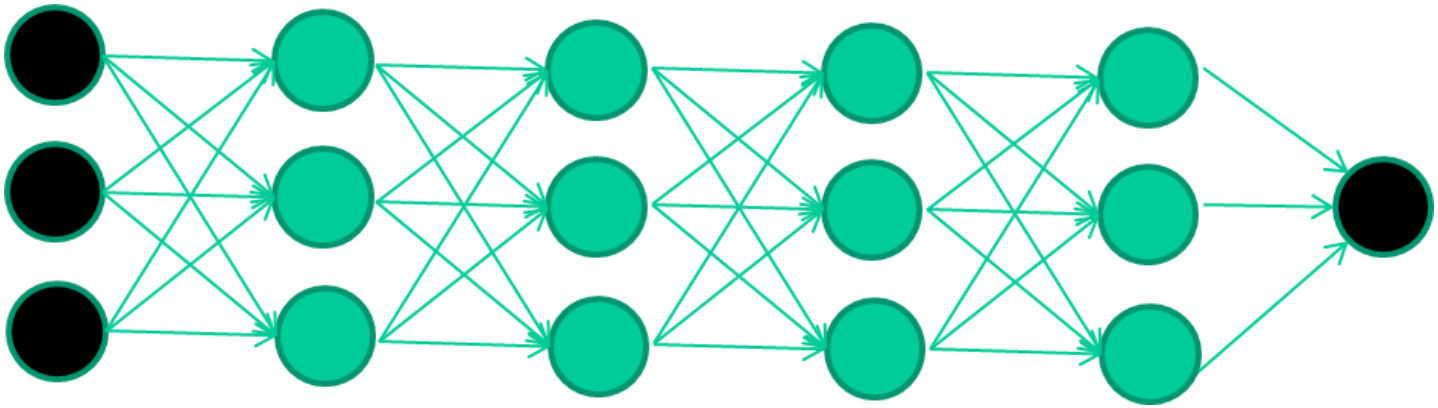
Tetapi, hingga saat ini, algoritma pembelajaran berat tidak berfungsi pada arsitektur multi-layer



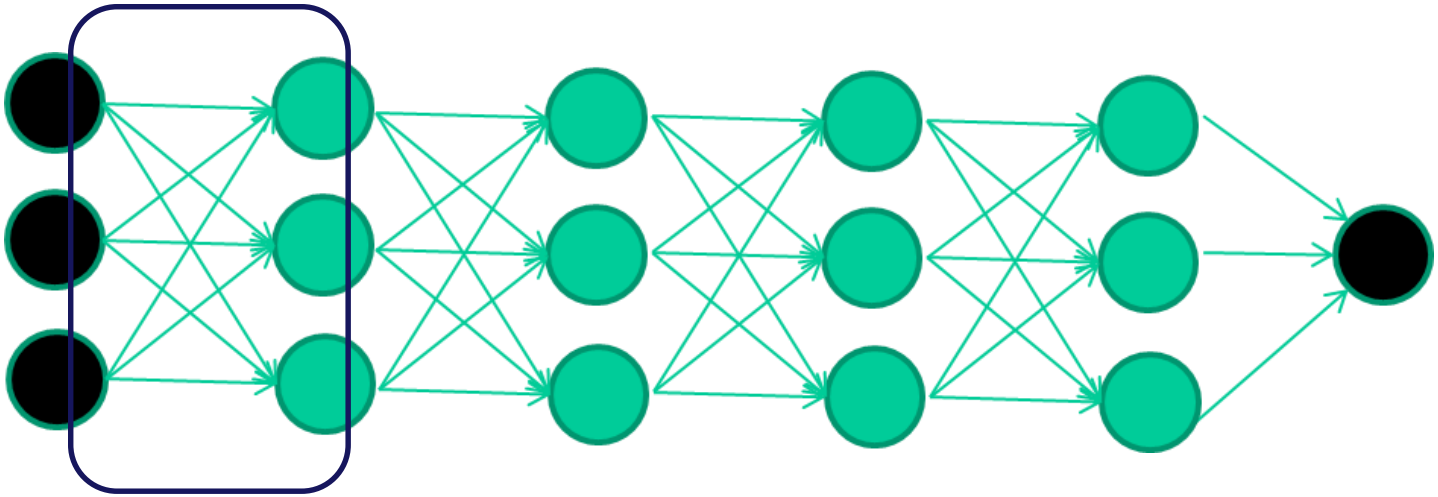


Kemudian datanglah “Deep Learning”

The new way to train multi-layer NNs…

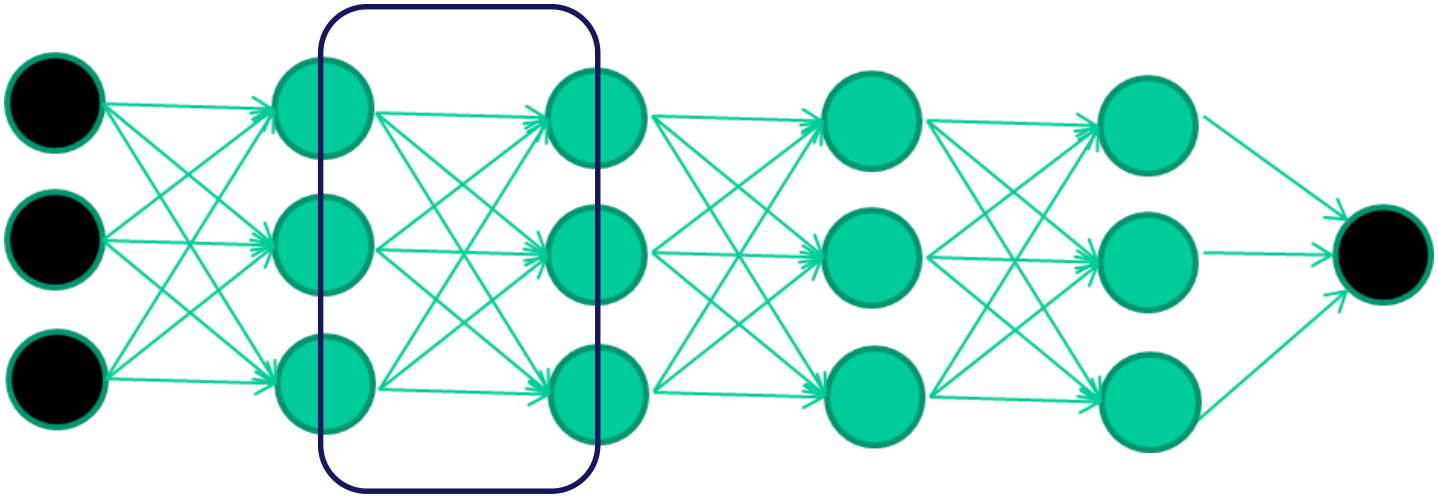


The new way to train multi-layer NNs…



Train **this** layer first

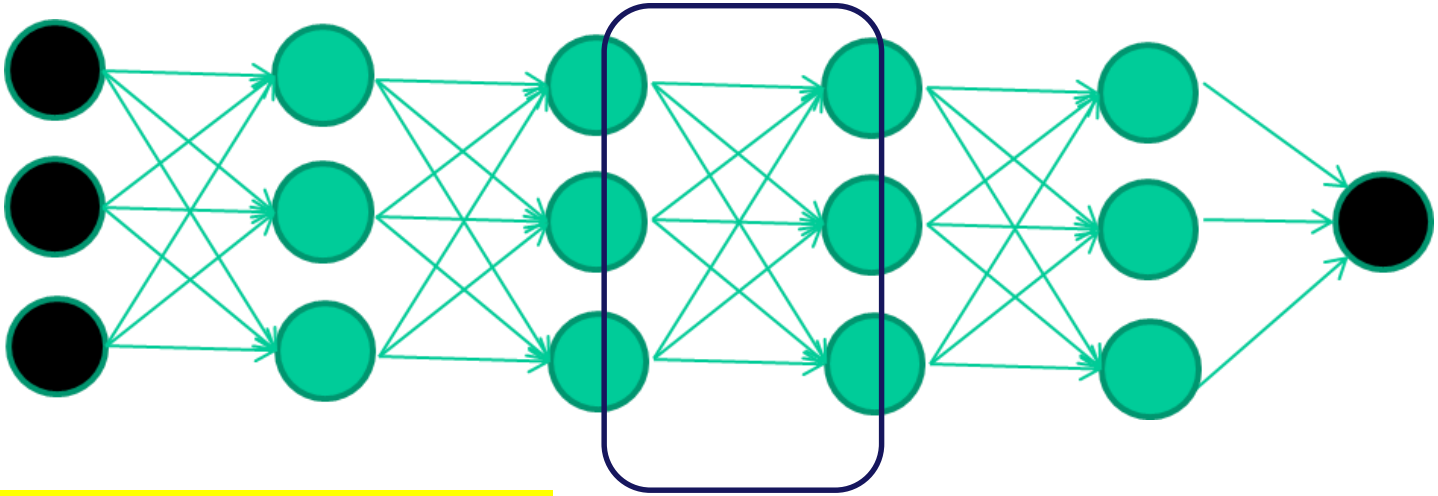
The new way to train multi-layer NNs…



Train **this** layer first

then **this** layer

The new way to train multi-layer NNs…

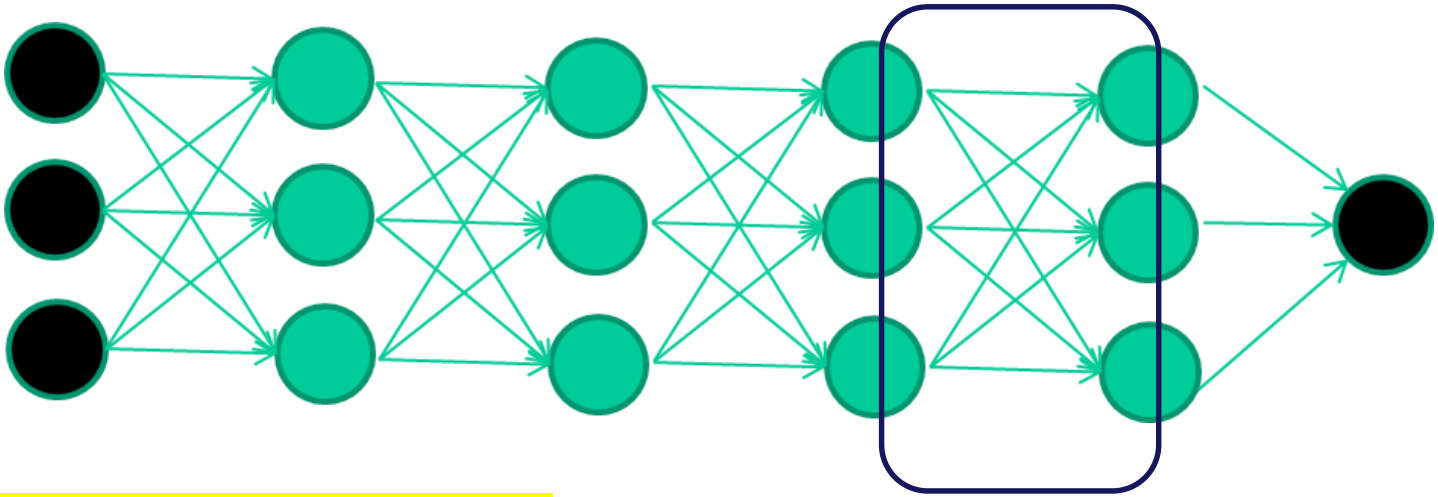


Train **this** layer first

then **this** layer

then **this** layer

The new way to train multi-layer NNs…



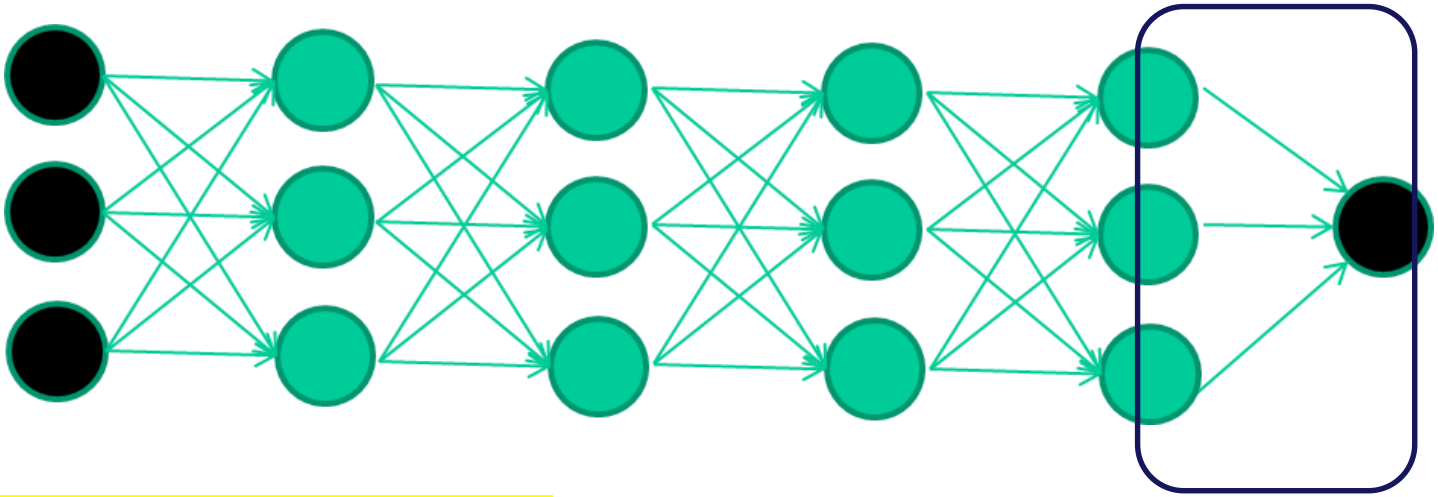
Train **this** layer first

then **this** layer

then **this** layer

then **this** layer

The new way to train multi-layer NNs…



Train **this** layer first

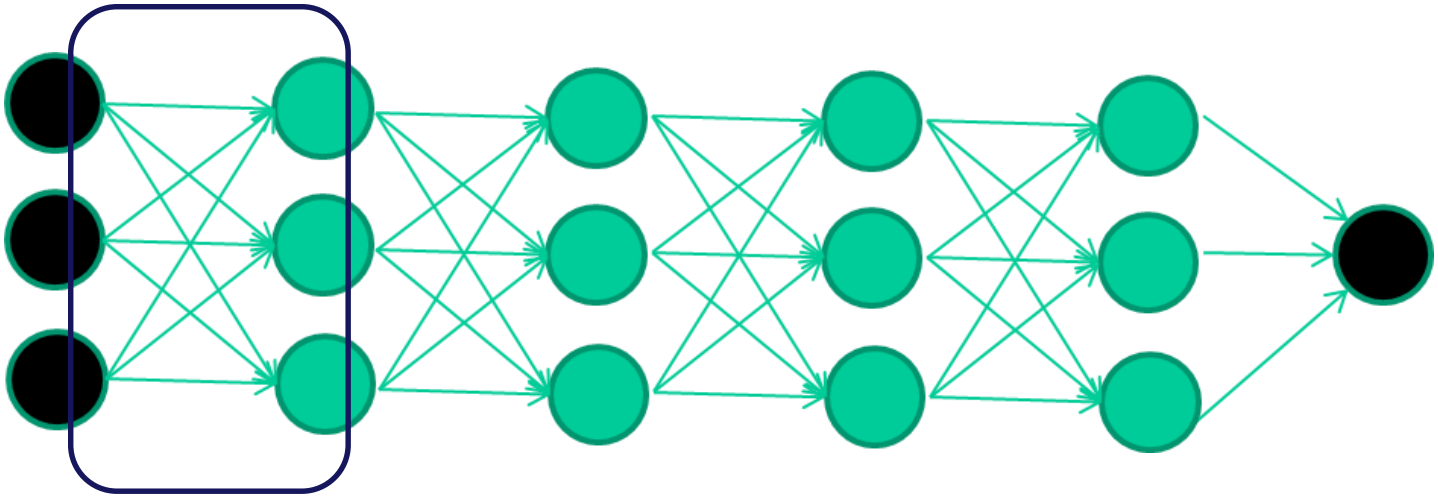
then **this** layer

then **this** laver

then **this** laver

finally **this** layer

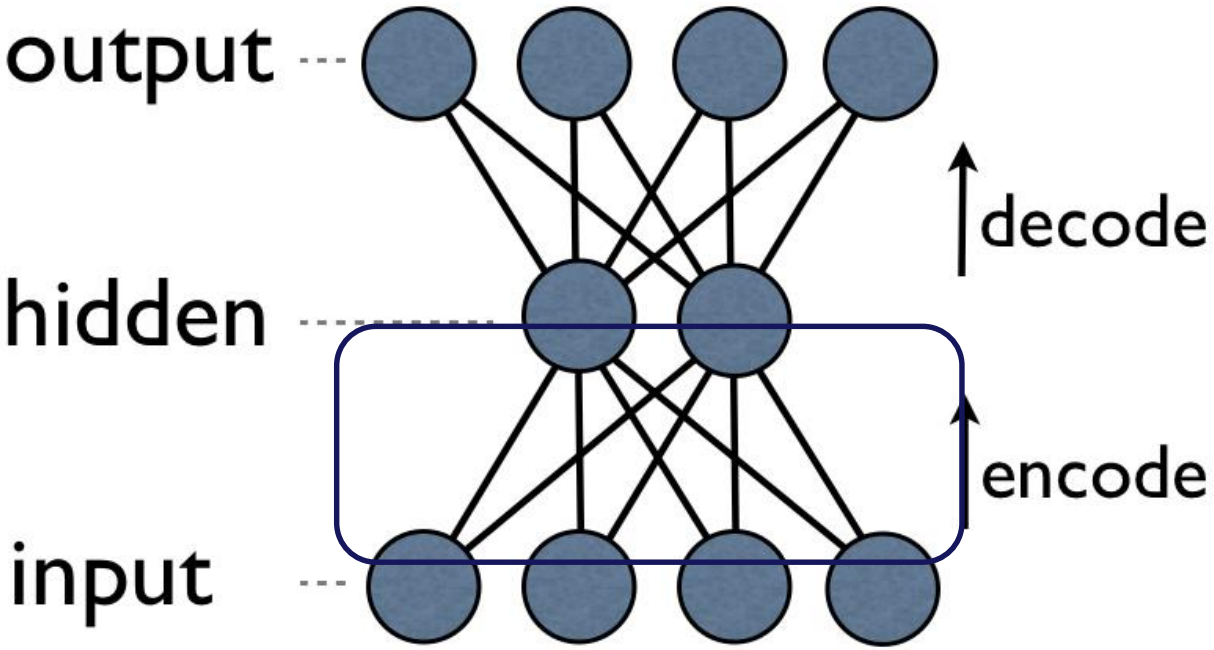
The new way to train multi-layer NNs…



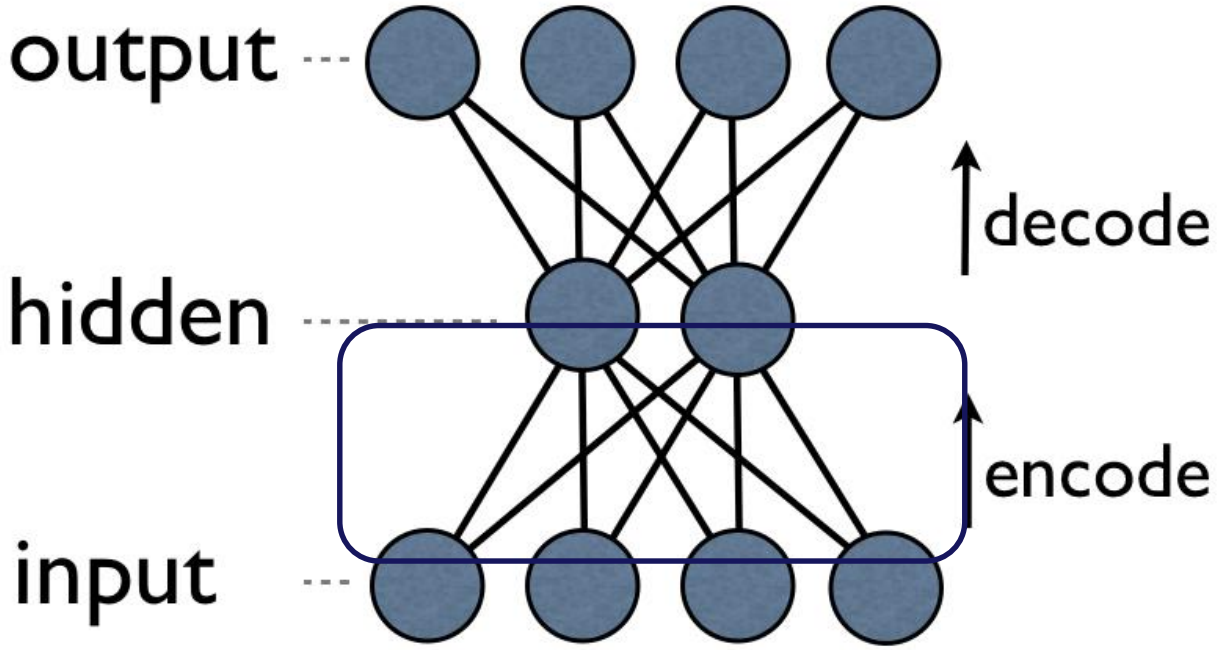
SETIAP lapisan (non-output) dilatih untuk menjadi auto-encoder

Pada dasarnya, ia dipaksa untuk mempelajari fitur-fitur bagus yang menggambarkan apa yang berasal dari lapisan sebelumnya

sebuah auto-encoder dilatih, dengan algoritma penyesuaian berat yang benar-benar standar untuk mereproduksi input

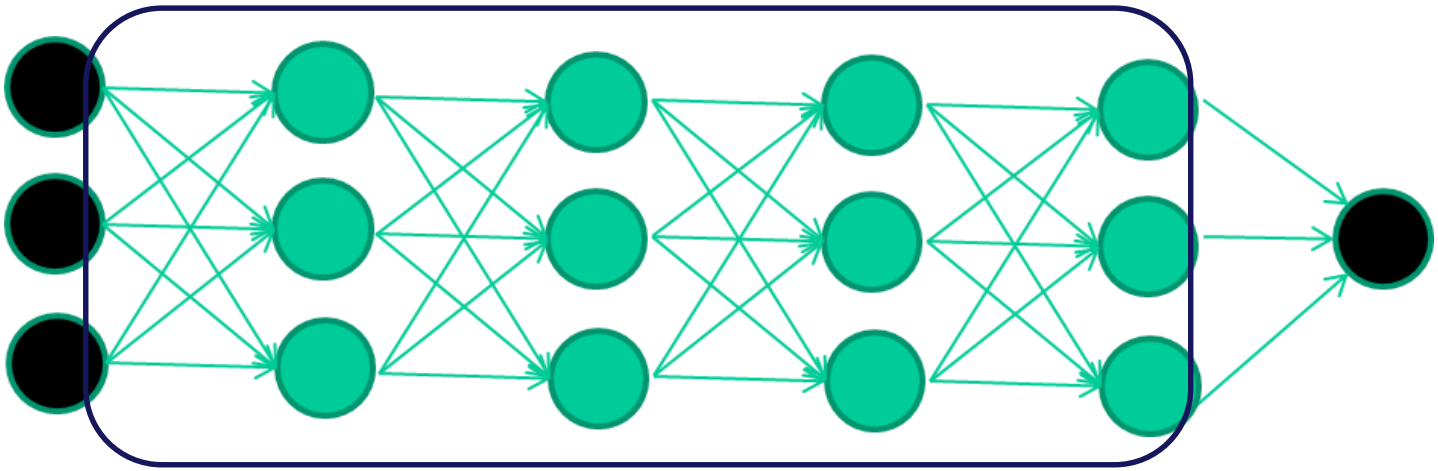


sebuah auto-encoder dilatih, dengan algoritma penyesuaian berat yang benar-benar standar untuk mereproduksi input

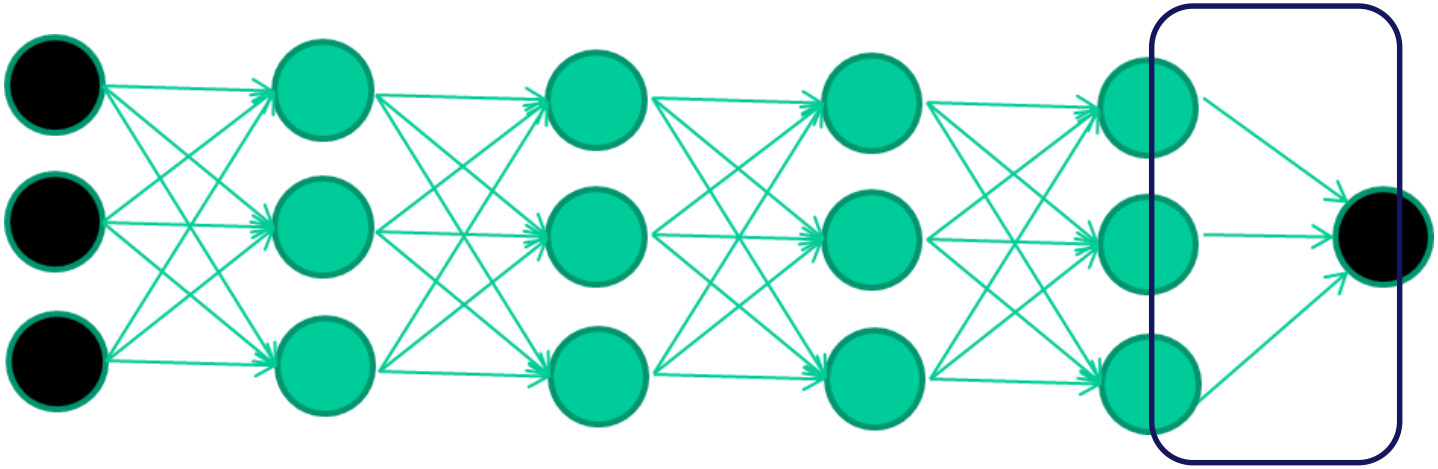


Dengan mewujudkan hal ini dengan (banyak) unit lebih sedikit daripada input, ini memaksa unit 'lapisan tersembunyi' untuk menjadi pendeteksi fitur yang baik

lapisan menengah masing-masing dilatih untuk menjadi penyandi otomatis (atau serupa)



Lapisan akhir dilatih untuk memprediksi kelas berdasarkan keluaran dari lapisan sebelumnya



Penutup

- Ada banyak banyak jenis pembelajaran mendalam (*Deep Learning*), berbagai jenis autoencoder, variasi arsitektur dan algoritma pelatihan, dll ...
- Area penelitian yang tumbuh sangat cepat