MachineLearning with Patterns Based on Lengyel-Epstein model

2017010698
수학과 오서영

1. Chemical pre-pattern and reaction-diffusion models for pigmentation

System of reacting and diffusing morphogens 3차원 구조 형성물질
could generate a chemical pre-pattern
within the developing integument 외피 via Turing instability

steady state
stable – diffusion X
unstable – diffusion O

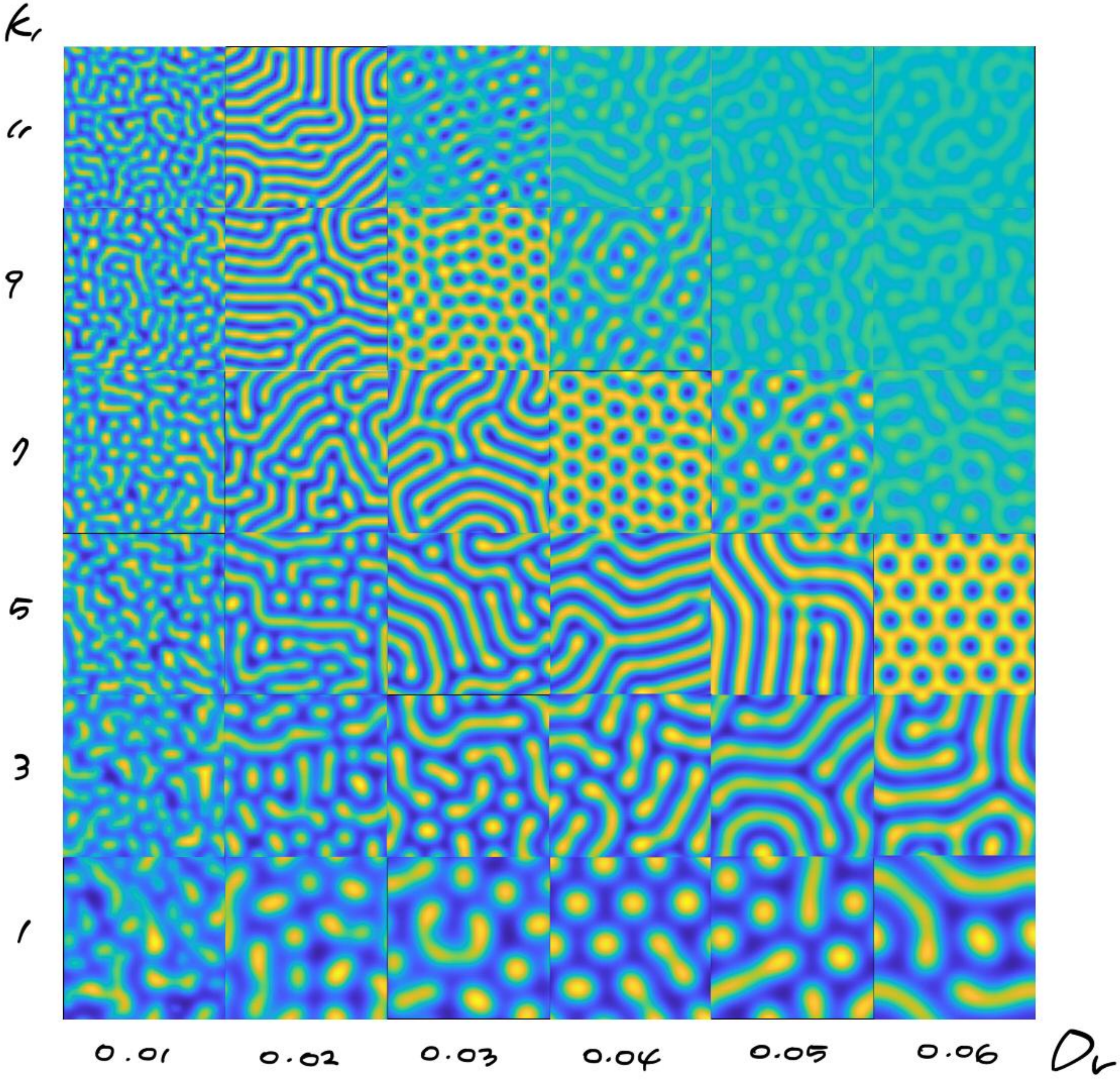
$$\frac{\partial u}{\partial t} = D_u \nabla^2 u + k_1 \left(v - \frac{uv}{1+v^2} \right)$$

$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + k_2 - v - \frac{4uv}{1+v^2},$$

2. Creating the pattern images(2D) based on Lengyel-Epstein model with MATLAB
: To classify 3 dissimilar patterns through a Neural Network

```
% set the initial condition
u=ubar+0.1*(2*rand(nx+2,ny+2)-1);
v=vbar+0.1*(2*rand(nx+2,ny+2)-1);
nu=u; nv=v;
```

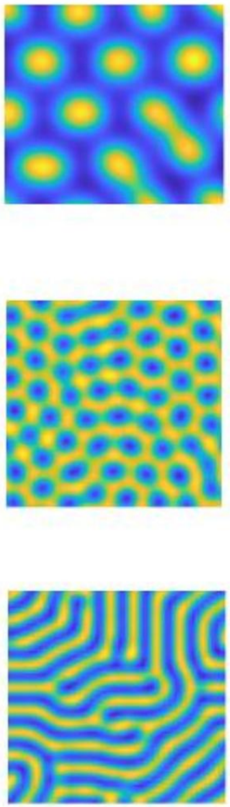
random
perturbation



k_1
11
9
7
5
3
1
0.01
0.02
0.03
0.04
0.05
0.06
D_r

3. Gradient Descent

input	output
X	Y



— 1

— 2

— 3

"random"
perturbation
↓
각각 1000개의
data

$$\left\{ \left(\underset{\text{image}}{x^{(1)}}, \overset{①\sim③}{y^{(1)}}\right), \left(x^{(2)}, y^{(2)}\right), \cdots, \left(x^{(3000)}, y^{(3000)}\right) \right\}$$

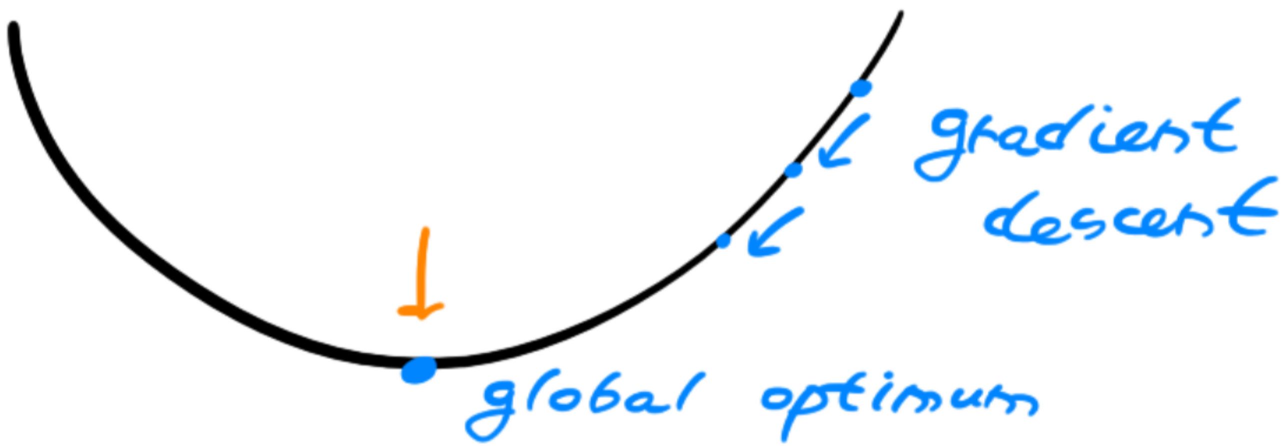
training test

$$x \rightarrow z = \underbrace{w x + b}_{\text{training}} \rightarrow g(z) \rightarrow \hat{y}$$

Gradient Descent

: want to find w, b
that minimize J

→ J(w, b) : convex



4. Single-Layer Neural Network with Softmax

5. CNN

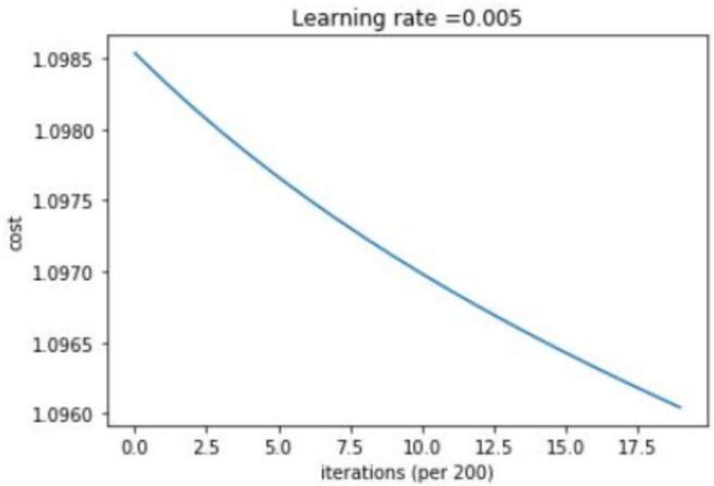
6. Gradient feature

7. ADAM Regularization

◦ 2000 iteration

gradient descent

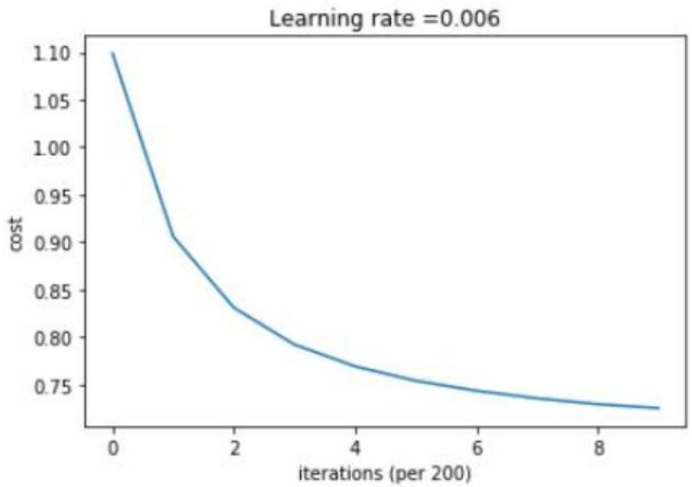
```
Cost after iteration 0: 1.098536
Cost after iteration 200: 1.098158
Cost after iteration 400: 1.097820
Cost after iteration 600: 1.097515
Cost after iteration 800: 1.097238
Cost after iteration 1000: 1.096985
Cost after iteration 1200: 1.096751
Cost after iteration 1400: 1.096533
Cost after iteration 1600: 1.096329
Cost after iteration 1800: 1.096136
```



train accuracy : 0.3341666666666667
test accuracy : 0.33

ADAM

```
Cost after iteration 0: 1.098627
Cost after iteration 200: 0.906004
Cost after iteration 400: 0.831171
Cost after iteration 600: 0.792285
Cost after iteration 800: 0.769404
Cost after iteration 1000: 0.754191
Cost after iteration 1200: 0.743681
Cost after iteration 1400: 0.735736
Cost after iteration 1600: 0.729602
Cost after iteration 1800: 0.725414
```

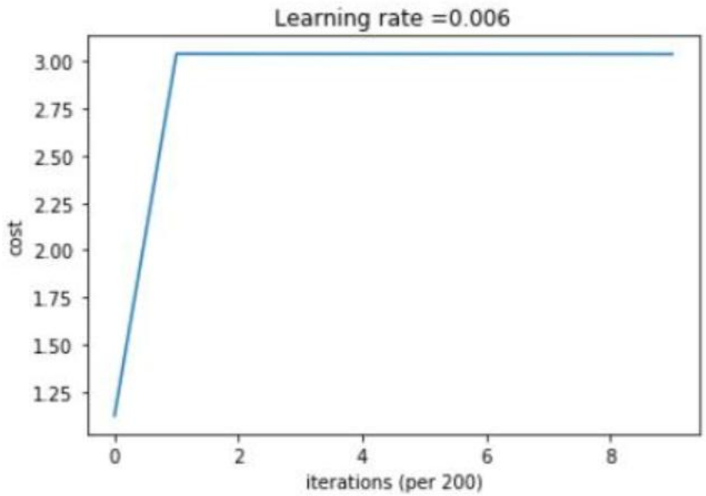


train accuracy : 0.86
test accuracy : 0.34

∘ 2000 iteration

non - gradient

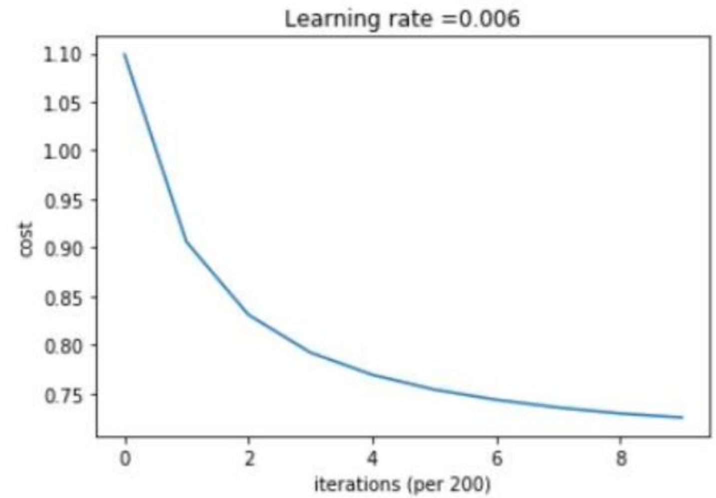
```
Cost after iteration 0: 1.123351
Cost after iteration 200: 3.039338
Cost after iteration 400: 3.039131
Cost after iteration 600: 3.038925
Cost after iteration 800: 3.038720
Cost after iteration 1000: 3.038515
Cost after iteration 1200: 3.038310
Cost after iteration 1400: 3.038107
Cost after iteration 1600: 3.037904
Cost after iteration 1800: 3.037701
```




```
train accuracy :  0.335
test accuracy :  0.32666666666666666
```

gradient

```
Cost after iteration 0: 1.098627
Cost after iteration 200: 0.906004
Cost after iteration 400: 0.831171
Cost after iteration 600: 0.792285
Cost after iteration 800: 0.769404
Cost after iteration 1000: 0.754191
Cost after iteration 1200: 0.743681
Cost after iteration 1400: 0.735736
Cost after iteration 1600: 0.729602
Cost after iteration 1800: 0.725414
```




```
train accuracy :  0.86
test accuracy :  0.34
```

8. U-U^3

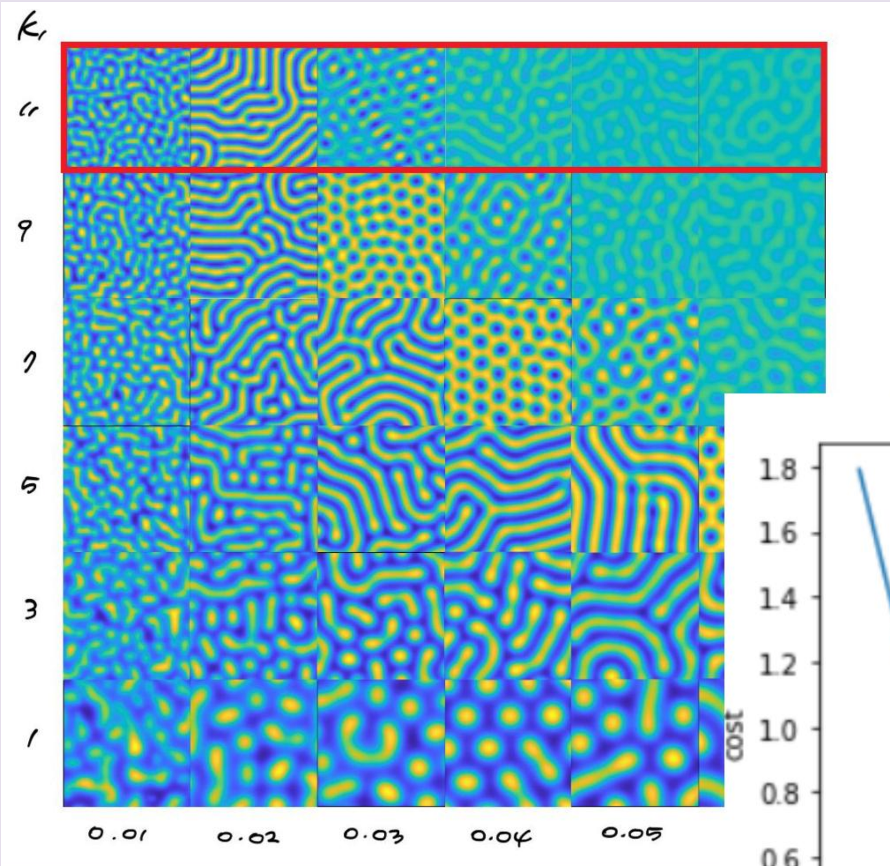
9. the number of training data can affect accuracy

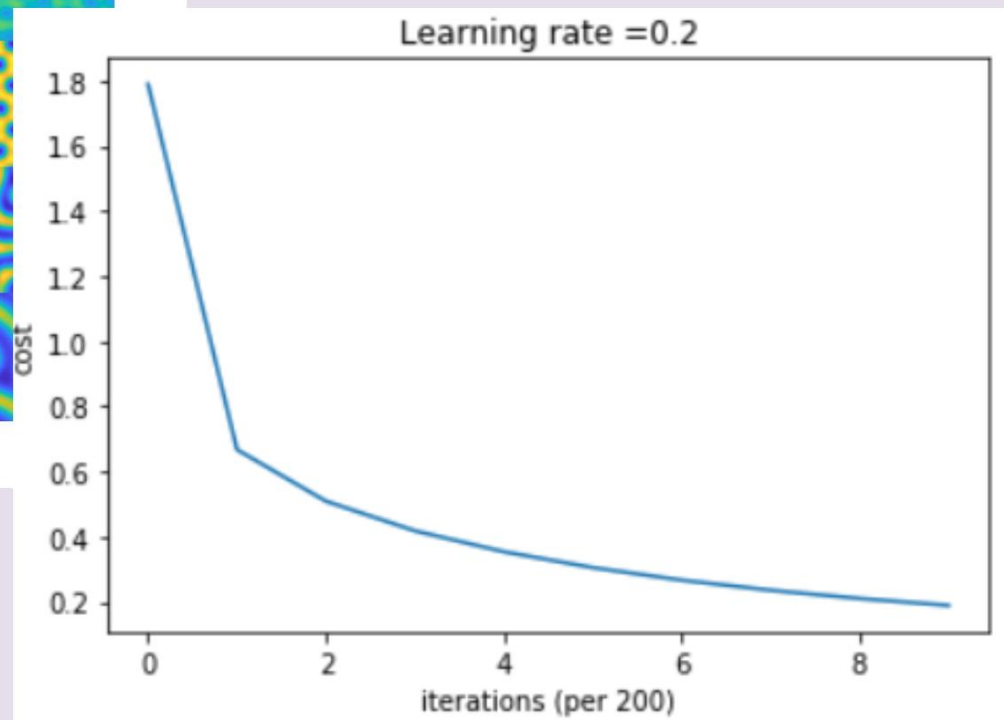
10. Compare 10 cases

Dissimilar	train :100 all : 144	train : 250 all : 360	train : 500 all : 720	train :1000 all : 1440
1) Cnn	100 / 1.0 1.0	100 / 1.0 1.0	100 / 1.0 1.0	100 / 1.0 1.0
2) 1-layer with GD	2000/ 0.99 0.005/ 0.43	2000/ 0.78 0.005/ 0.36	2000/ 0.67 0.004/ 0.36	2000/ 0.60 0.004/ 0.43
3) 1layer with Adam	2000/ 1.0 0.1/ 0.47	2000/ 1.0 0.1/ 0.40	2000/ 0.79 0.03/ 0.36	2000/ 0.71 0.01/ 0.43
4) Derivative layer with GD	2000/ 1.0 10/ 0.36	2000/ 1.0 10/ 0.37	2000/ 1.0 10/ 0.40	2000/ 0.96 10/ 0.34
5) Derivative layer with Adam	2000/ 1.0 1/ 0.31	2000/ 1.0 0.9/ 0.37	2000/ 1.0 0.9/ 0.36	2000/ 1.0 0.9/ 0.34
6) 2weight with GD	2000/ 1.0 0.005/ 0.43	2000/ 0.79 0.005/ 0.43	2000/ 0.68 0.004/ 0.42	2000/ 0.58 0.004/ 0.43
7) 2weight with Adam	2000/ 1.0 0.07/ 0.38	2000/ 1.0 0.06/ 0.41	2000/ 1.0 0.06/ 0.44	2000/ 0.96 0.06/ 0.40
8) 2weight & u-u^3 with GD	2000/ 1.0 0.001/ 0.77	2000/ 0.96 0.001/ 0.82	2000/ 0.95 0.001/ 0.90	2000/ 0.94 0.001/ 0.89
9) 2weight & u-u^3 with Adam	600/ 1.0 0.04/ 0.88	600/ 1.0 0.04/ 0.90	600/ 1.0 0.04/ 0.90	600/ 1.0 0.04/ 0.90

Similar	train :100 all : 144	train : 250 all : 360	train : 500 all : 720
1) Cnn	100 / 1.0 0.54	100 / 1.0 0.65	100 / 1.0 0.65
2) 1-layer with GD	2000/ 1.0 0.005/ 0.22	2000/ 0.96 0.005/ 0.34	2000/ 0.86 0.005/ 0.33
3) 1layer with Adam	2000/ 1.0 0.08/ 0.34	2000/ 1.0 0.08/ 0.36	2000/ 1.0 0.08/ 0.31
4) Derivative layer with GD	2000/ 1.0 10/ 0.36	2000/ 1.0 10/ 0.32	2000/ 1.0 10/ 0.37
5) Derivative layer with Adam	2000/ 1.0 1/ 0.34	2000/ 1.0 1/ 0.32	2000/ 1.0 1/ 0.42
6) 2weight with GD	2000/ 1.0 0.005/ 0.40	2000/ 0.97 0.005/ 0.33	2000/ 0.86 0.005/ 0.36
7) 2weight with Adam	2000/ 1.0 0.07/ 0.38	2000/ 1.0 0.07/ 0.37	2000/ 1.0 0.07/ 0.46
8) 2weight & u-u^3 with GD	2000/ 1.0 0.001/ 0.36	2000/ 0.97 0.001/ 0.24	2000/ 0.82 0.001/ 0.32
9) 2weight & u-u^3 with Adam	2000/ 1.0 0.04/ 0.31	2000/ 1.0 0.04/ 0.37	2000/ 1.0 0.04/ 0.37

11. Classification of all patterns






```
train accuracy :  1.0
test accuracy :  0.30092592592592593
```

12. Complex Pattern in a simple system
: Pearson's Classification of Gray-Scott System Parameter Values

Reaction - Diffusion equations

$$\frac{\partial U}{\partial t} = D_u \nabla^2 U - UV^2 + F(1 - U)$$

$$\frac{\partial V}{\partial t} = D_v \nabla^2 V + UV^2 - (F + k)V$$

The Two Parameters *F* and *k*

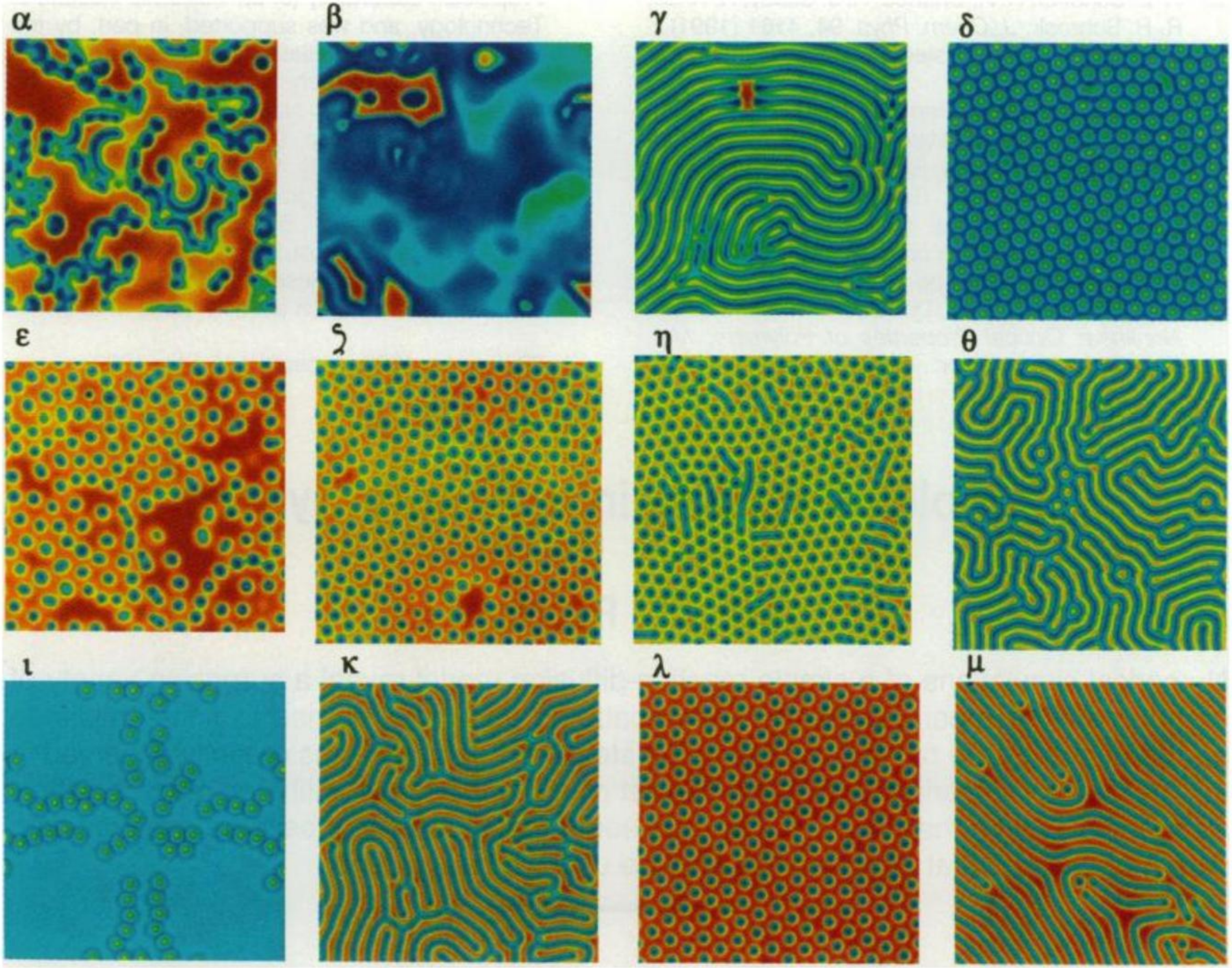
: Describes the parameters *F* and *k* as "feed rate" and "kill rate".

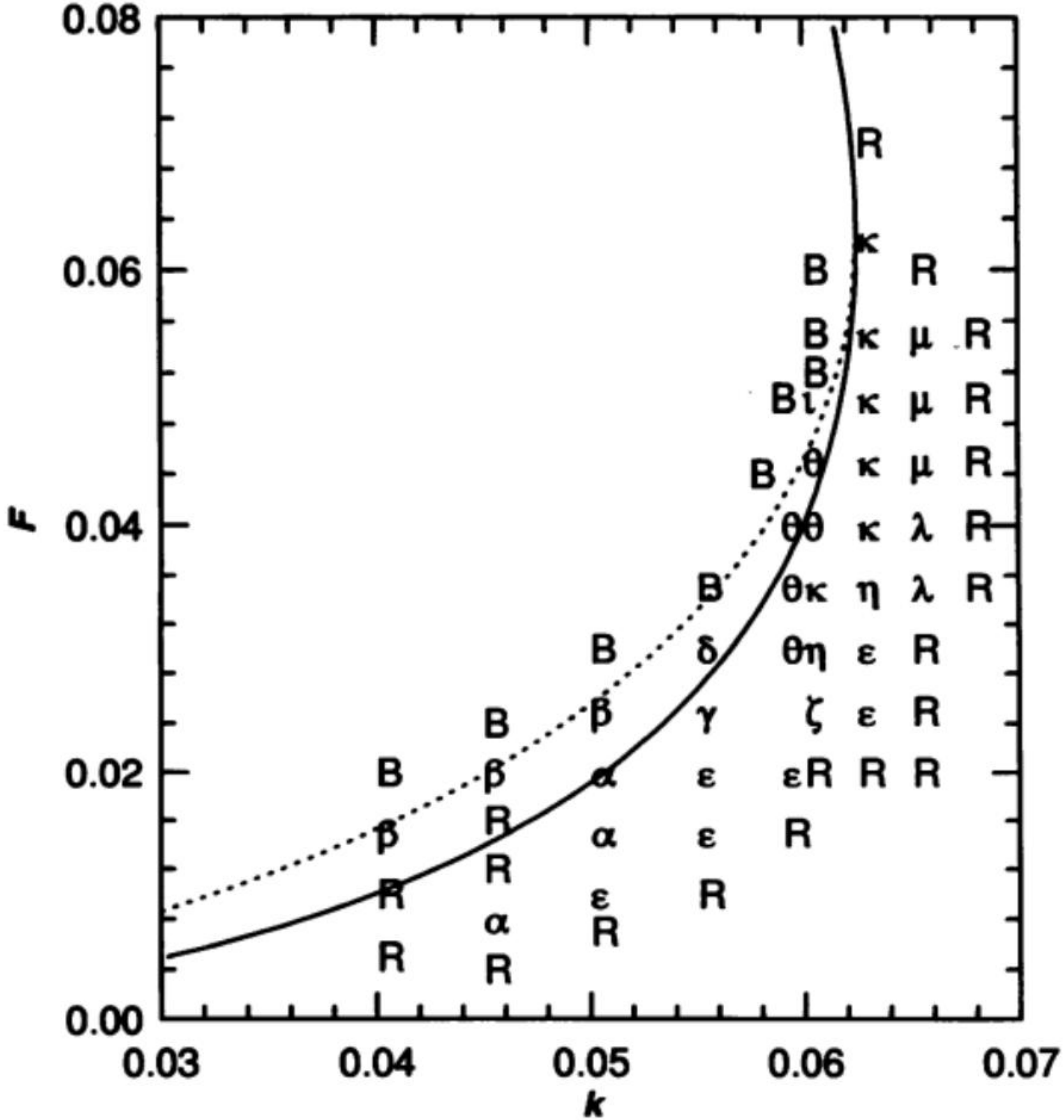
Feed-Rate Spectrum: Stability/Oscillation/Chaos

Varying the parameter F while keeping k constant causes the system to move vertically
in the parameter map at the top of this page.
Varying k in such a way as to follow the "crescent-shaped"
contour will reveal a spectrum of phenomena distinguished
by the amount and type, if any, of oscillation.

Kill-Rate Spectrum: Soliton Type and Shape

Varying just the parameter k while keeping F constant reveals a second spectrum
distinguished by the presence or absence of large solid regions, stripes, and/or spots.
This spectrum does not appear in the lower F values where the system is too chaotic.

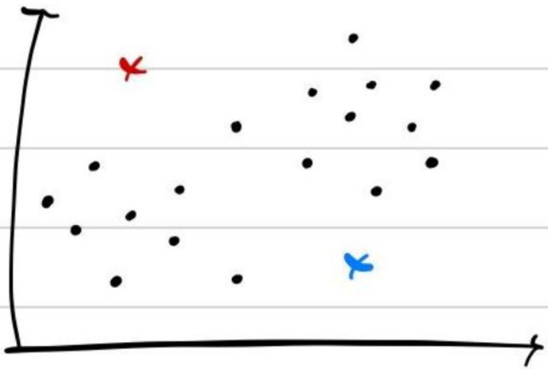

α
β
γ
δ
ε
ζ
η
θ
ι
κ
λ
μ



0.08
0.06
0.04
0.02
0.00
F
0.03
0.04
0.05
0.06
0.07
k

12. Clustering - k-means

• K - means algorithm



x, x : clustering centroids

① Clustering Assignment Step

: It's going to assign each of the data points one of the two cluster cluster centroids (depending on whether it is closer to them)

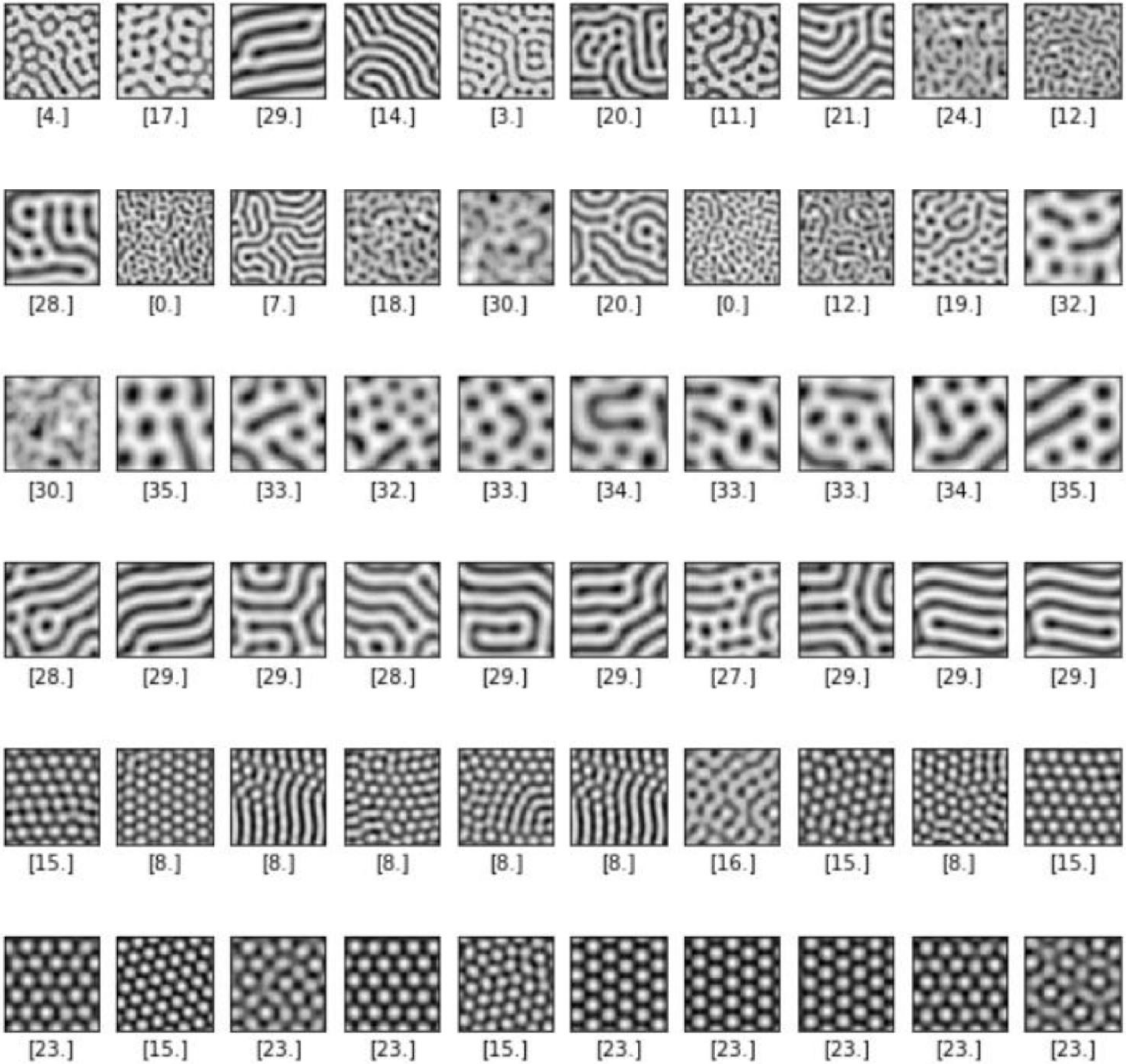
red on blue

② Move Centroid Step

: We are going to move two centroids to the average of the points colored the same colour. blue centroid $\xrightarrow{\text{move}}$ blue dot's mean

③ Go to back ① (depending on whether it's closer red or blue)

: If cluster centroids do not change and the colours of the points will not change then at this point k-means has converged


[4.] [17.] [29.] [14.] [3.] [20.] [11.] [21.] [24.] [12.]
[28.] [0.] [7.] [18.] [30.] [20.] [0.] [12.] [19.] [32.]
[30.] [35.] [33.] [32.] [33.] [34.] [33.] [33.] [34.] [35.]
[28.] [29.] [29.] [28.] [29.] [29.] [27.] [29.] [29.] [29.]
[15.] [8.] [8.] [8.] [8.] [8.] [16.] [15.] [8.] [15.]
[23.] [15.] [23.] [23.] [15.] [23.] [23.] [23.] [23.] [23.]