Microsoft

Azure Developer Tour

#azuredevtour

aka.ms/devtourcode

@azureadvocates

Azure

Productive + Hybrid + Intelligent + Trusted

Demo: A Lap Around Azure

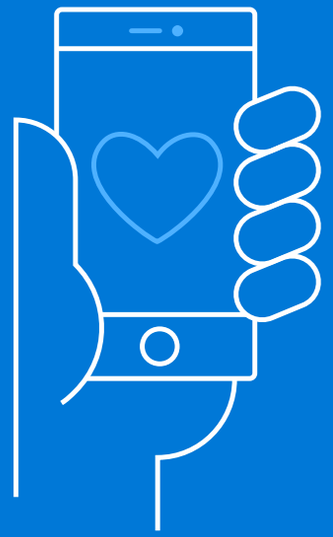
Web Apps

DevOps CI/CD

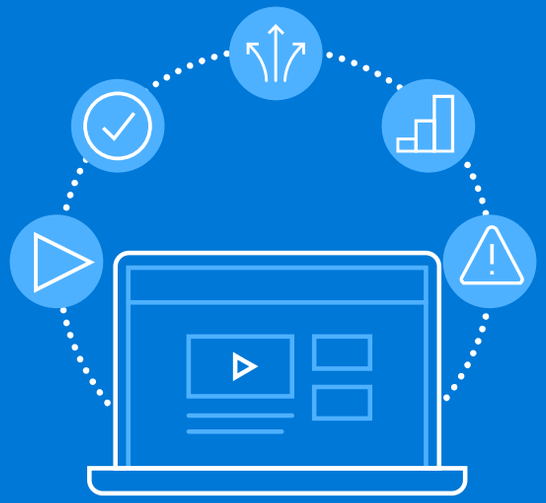
Security

Mobile

Essentials for mobile success



Beautiful native apps



Build, test, distribute, learn

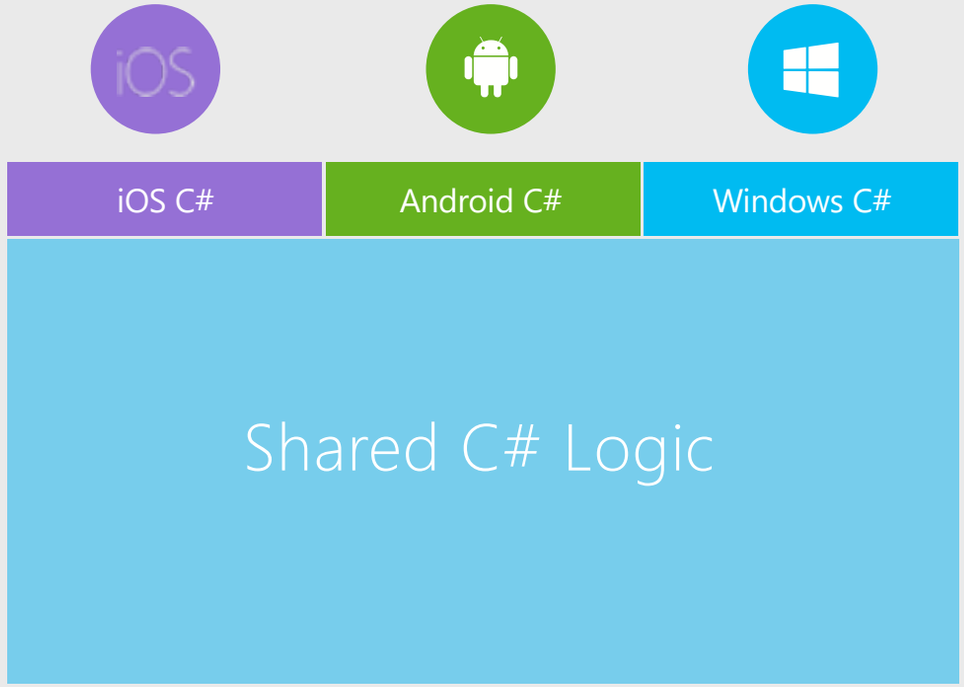
Intelligent cloud

+

Azure

The Best Way to Build Cloud-Connected Mobile Apps

Xamarin Native – Traditional UI



- 3 Native User Interfaces
- Shared App Logic

Xamarin.Forms- Cross Platform UI

- Shared Native User Interface
- Shared App Logic

Demo: Xamarin

Containers

Deploy containers everywhere in Azure

Azure Container Instances

Easily deploy and run containers with a single command

Launch container instances in seconds

Cost effective per second billing

Azure Container Service (AKS)

Fully managed Kubernetes orchestration service

Auto patching, auto scaling, auto updates

Use the full Kubernetes ecosystem

Azure and Azure Stack

The latest containers news

- ACI: Generally Available
- AKS: Generally Available very soon!
- Azure DevOps Project
- VS Code: Debug in-cluster containers using Draft Open Service Broker for Azure
- Azure Dev Spaces

Serverless

No infrastructure management

Pay for what you use

Infinite scale

Focus on your code

Serverless

Azure Functions

Powerful triggers and bindings

HTTP as a first class citizen

Best-in-class tooling

Open source

Pay per execution

AI and Machine Learning

data
science

machine learning

- finding (and exploiting) patterns in data
- replacing “human writing code” with “human supplying data”
  - system figures out what the person wants based on examples
  - need to abstract from “training” examples to “test” examples
  - most central issue in ML: generalization
- starts with a *sharp* question

machine learning

- how much / how many
- which class does this belong to?
- are there different groups? which does it belong to?
- is this weird?
- which option should I choose?

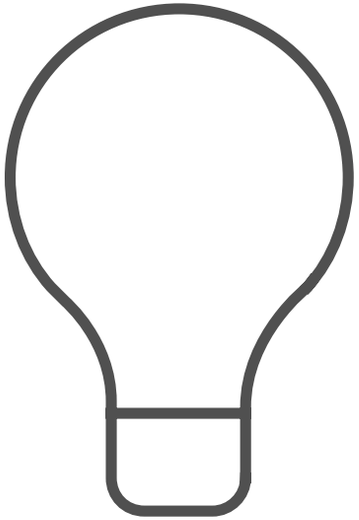
machine learning

supervised

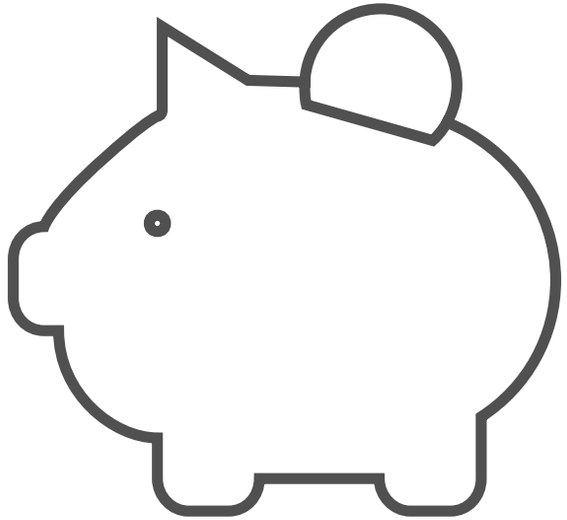
- (regression)
  how much / how many
- (classification)
  which class does this belong to?

unsupervised

- (clustering)
  are there different groups? which does it belong to?
- (anomaly detection)
  is this weird?
- (recommendation)
  which option should I choose?



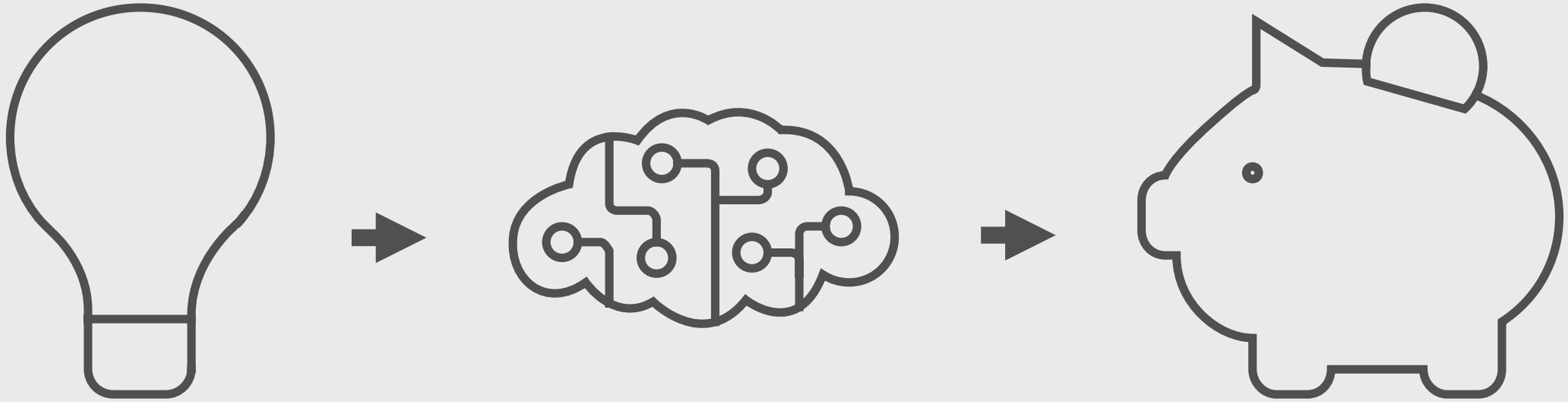
?



idea

profit

option 1 – easy



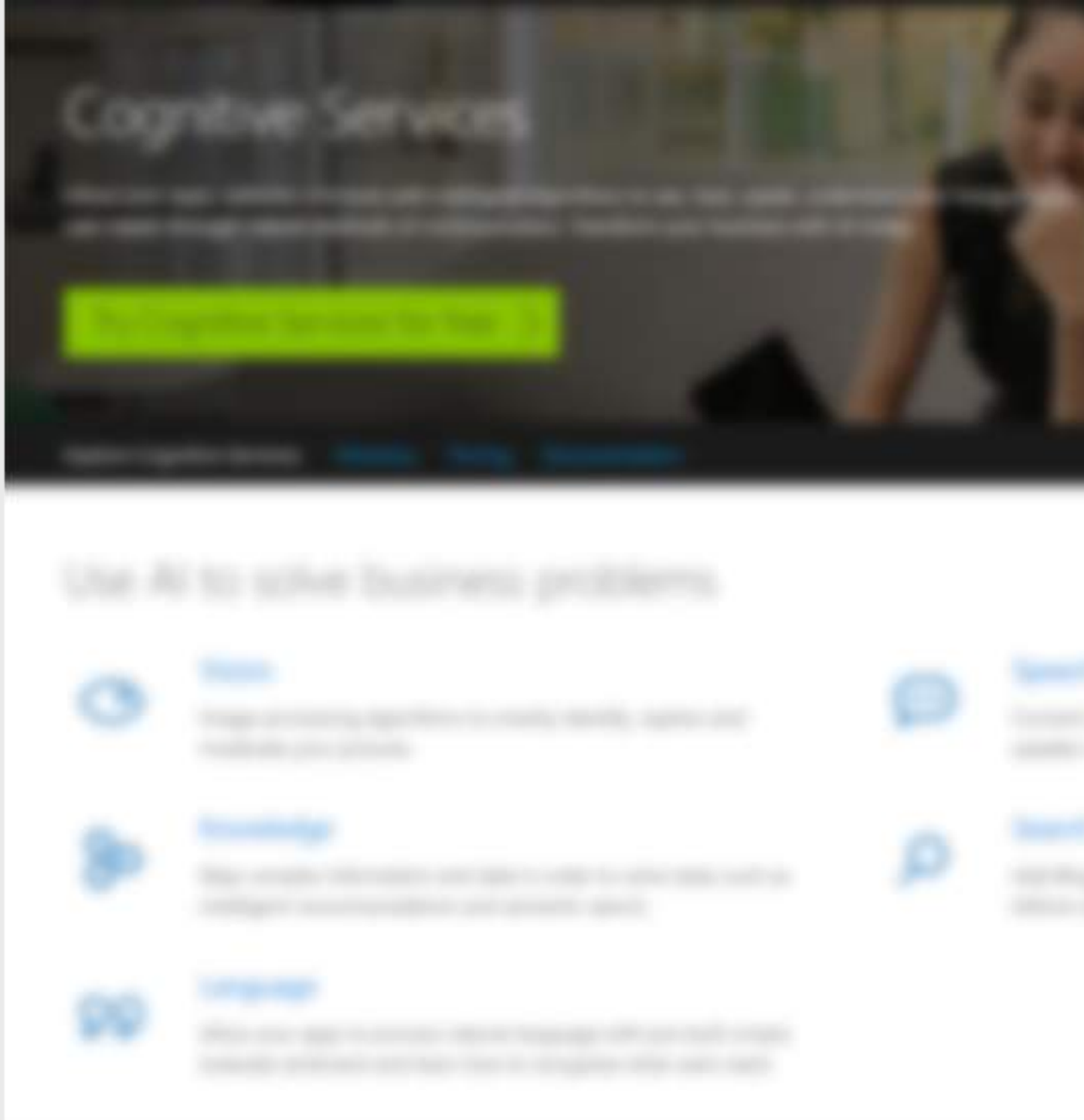
idea

prebuilt AI

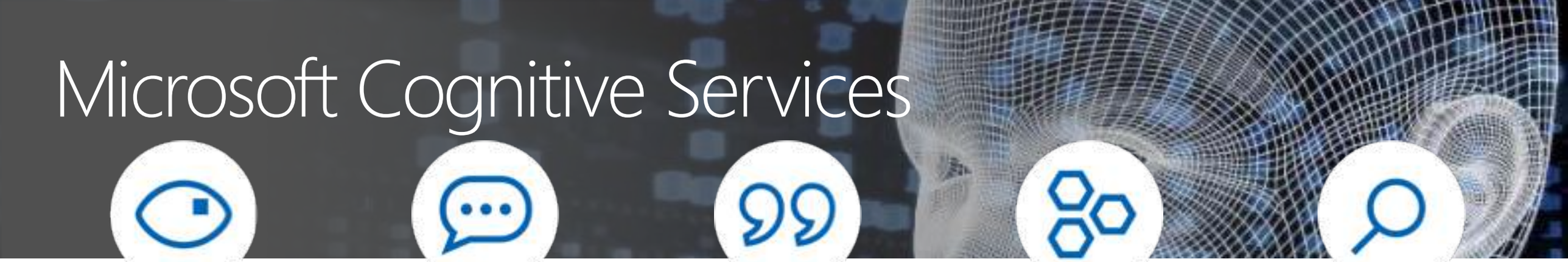
profit

cognitive services

Infuse your apps, websites and bots with intelligent algorithms to see, hear, speak, understand and interpret your user needs through natural methods of communication



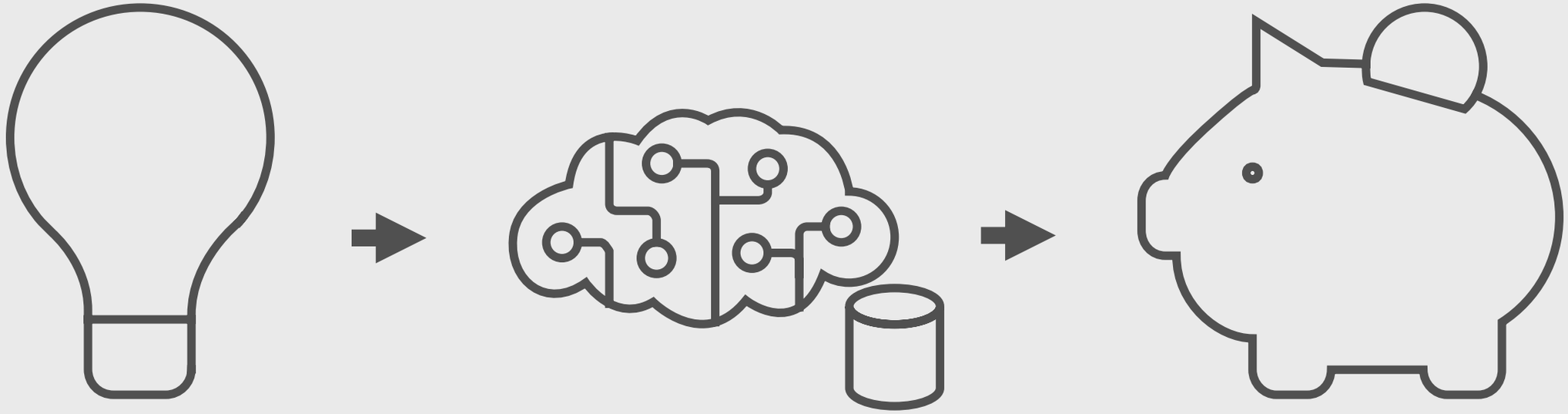
Microsoft Cognitive Services



Vision	Speech	Language	Knowledge	Search
Video Indexer	Speaker Recognition	Text Analytics	QnA Maker	Bing Entity Search
Computer Vision	Bing Speech	Bing Spell Check		Bing Autosuggest
Face	Translator Speech	Translator Text		Bing Search
Emotion	Unified Speech			Bing Statistics add-in
Content Moderator				Bing Visual Search

Demo: Cognitive Services

option 2 – still easy, but more work

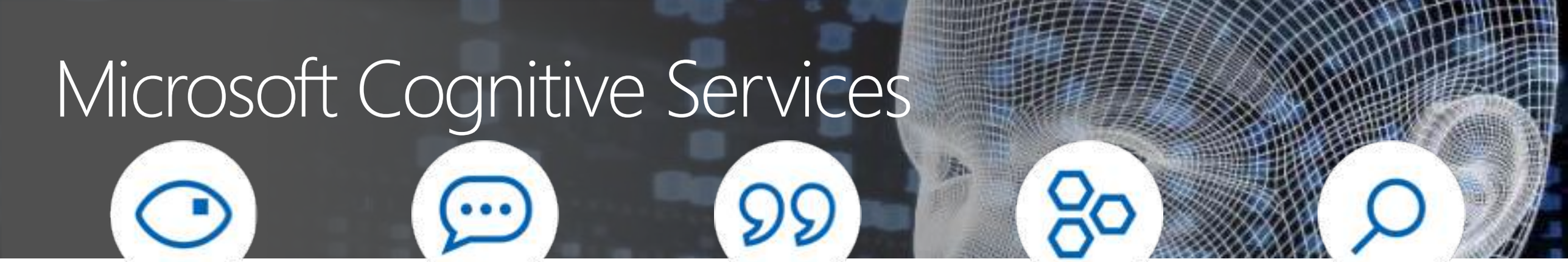


idea

customize
prebuilt AI

profit

Microsoft Cognitive Services



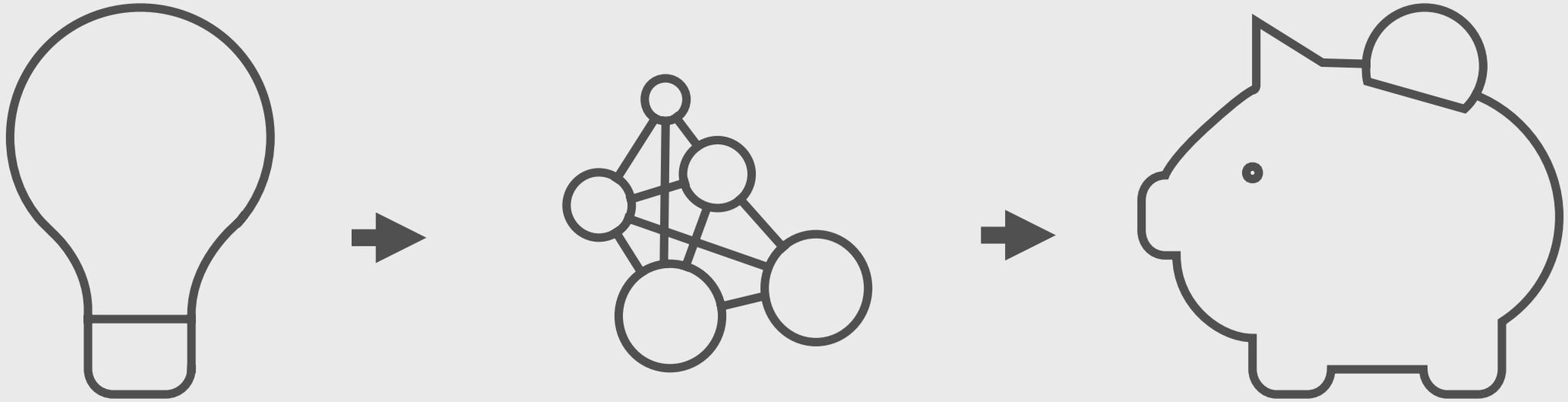
Vision	Speech	Language	Knowledge	Search
Video Indexer	Speaker Recognition	Text Analytics	QnA Maker	Bing Entity Search
Computer Vision	Bing Speech	Bing Spell Check		Bing Autosuggest
Face	Translator Speech	Translator Text		Bing Search
Emotion	Unified Speech			Bing Statistics add-in
Content Moderator				Bing Visual Search

Customization

Custom Vision	Custom Speech	Language Understanding (LUIS)	Custom Decision	Bing Custom Search

Demo: Cognitive Services, Pt 2

option 3 – hands will be dirtied

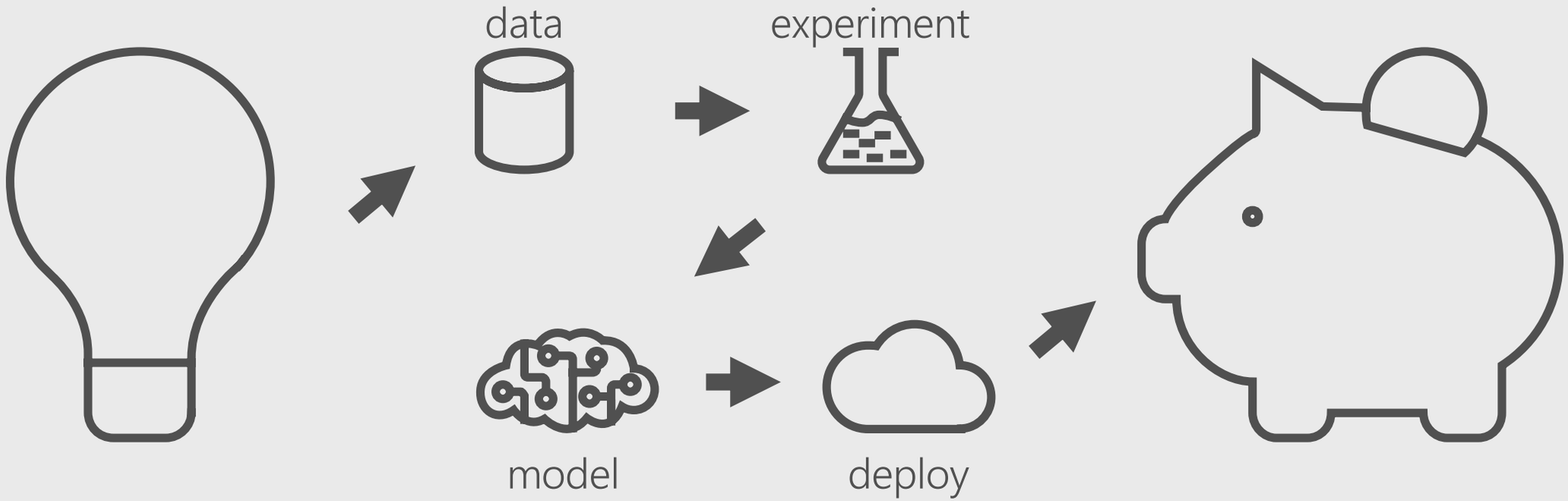


idea

custom AI

profit

option 3



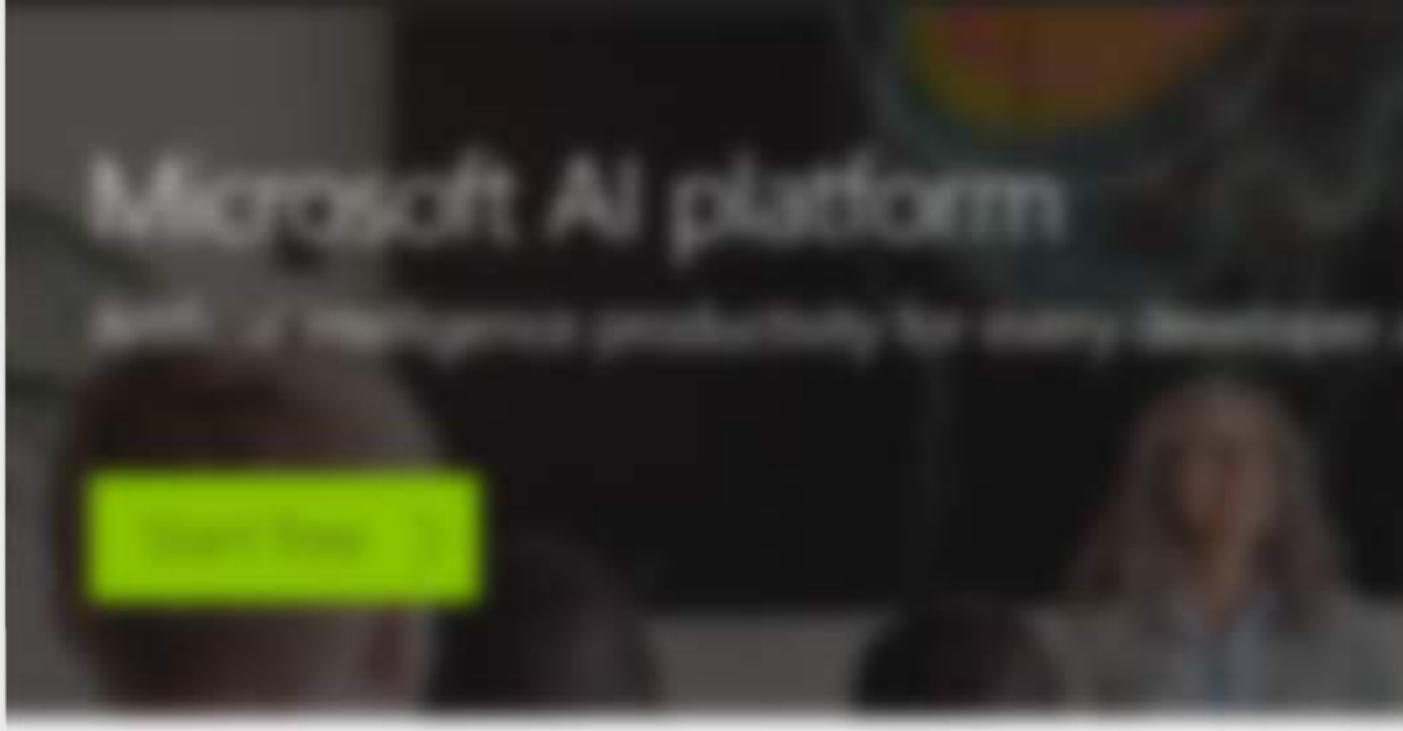
idea

custom AI

profit

AI platform

Artificial Intelligence productivity for every developer and every scenario

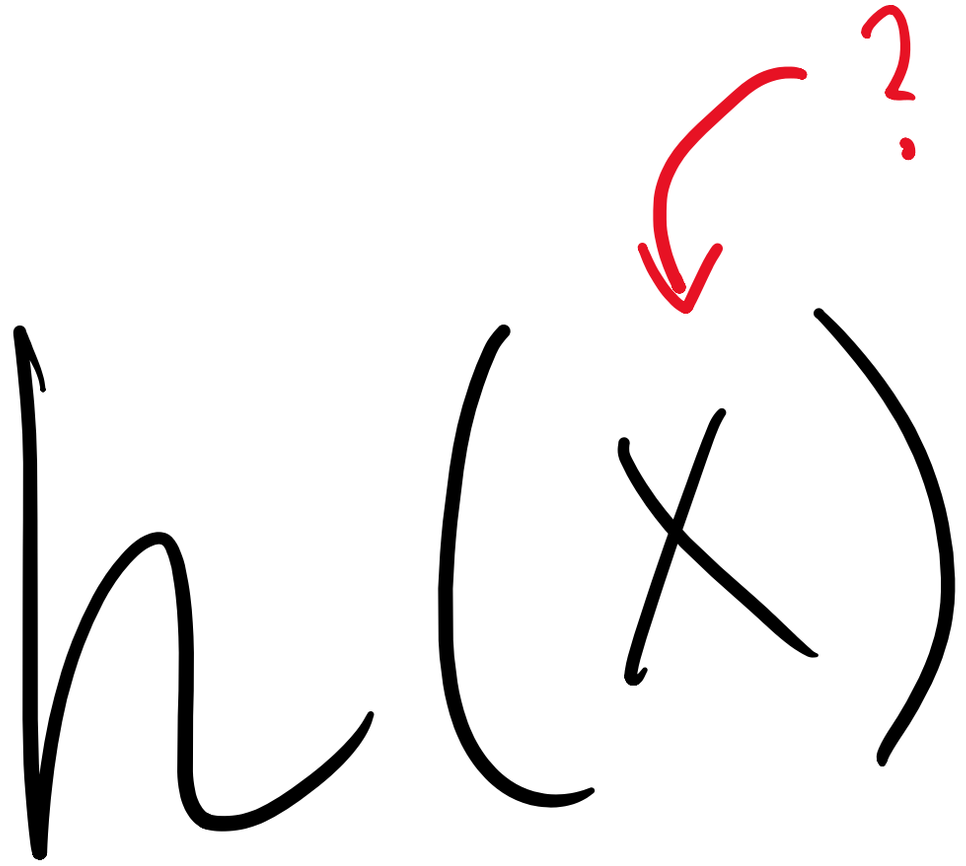


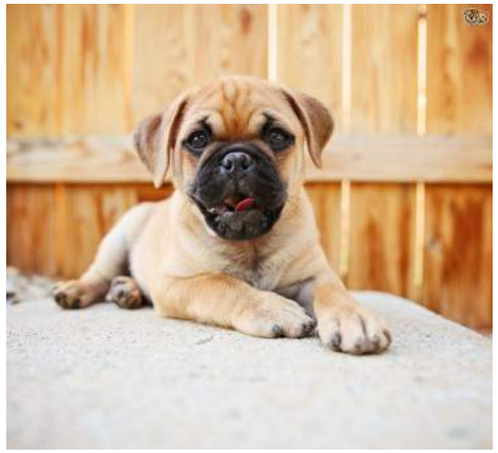
Demo: Deep Learning

(float, float)

getCoolScores(picture)

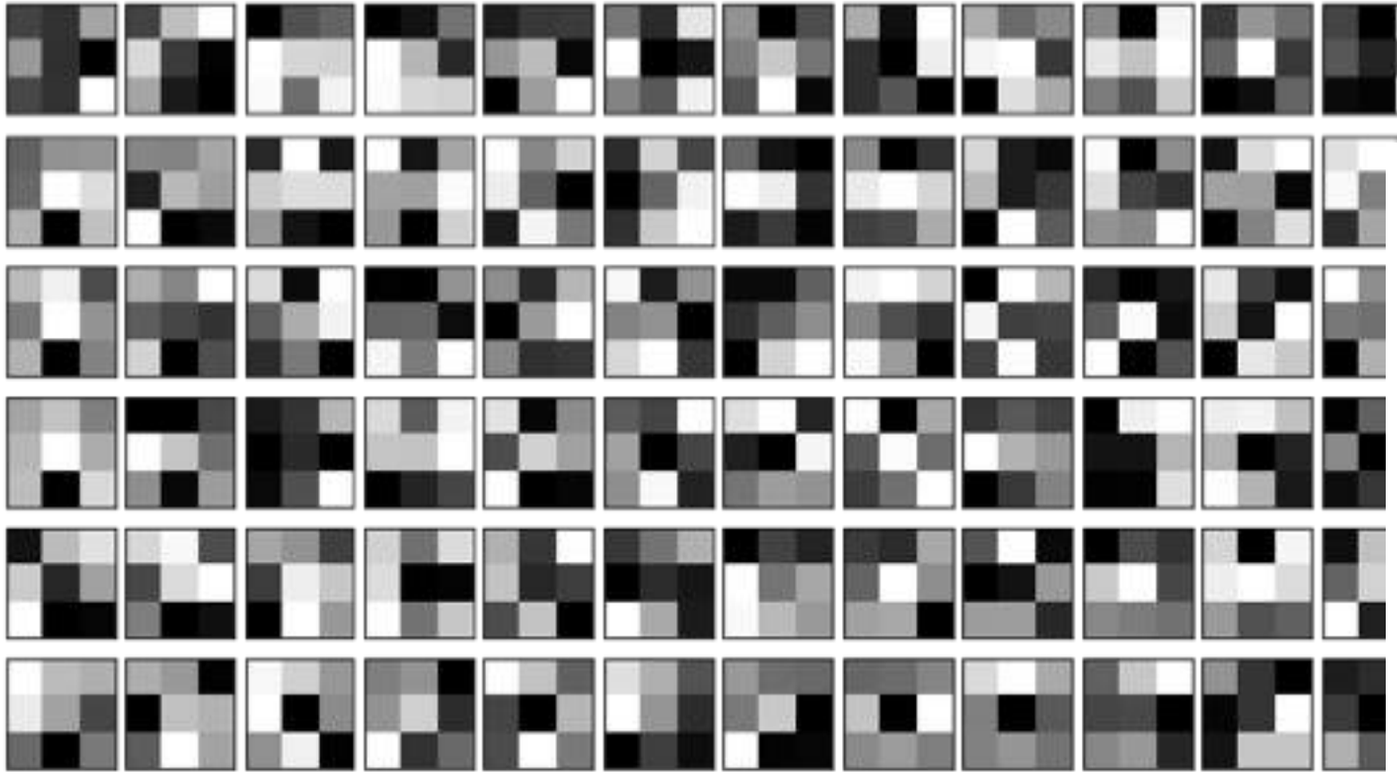
$h(x)$


?
h(x)

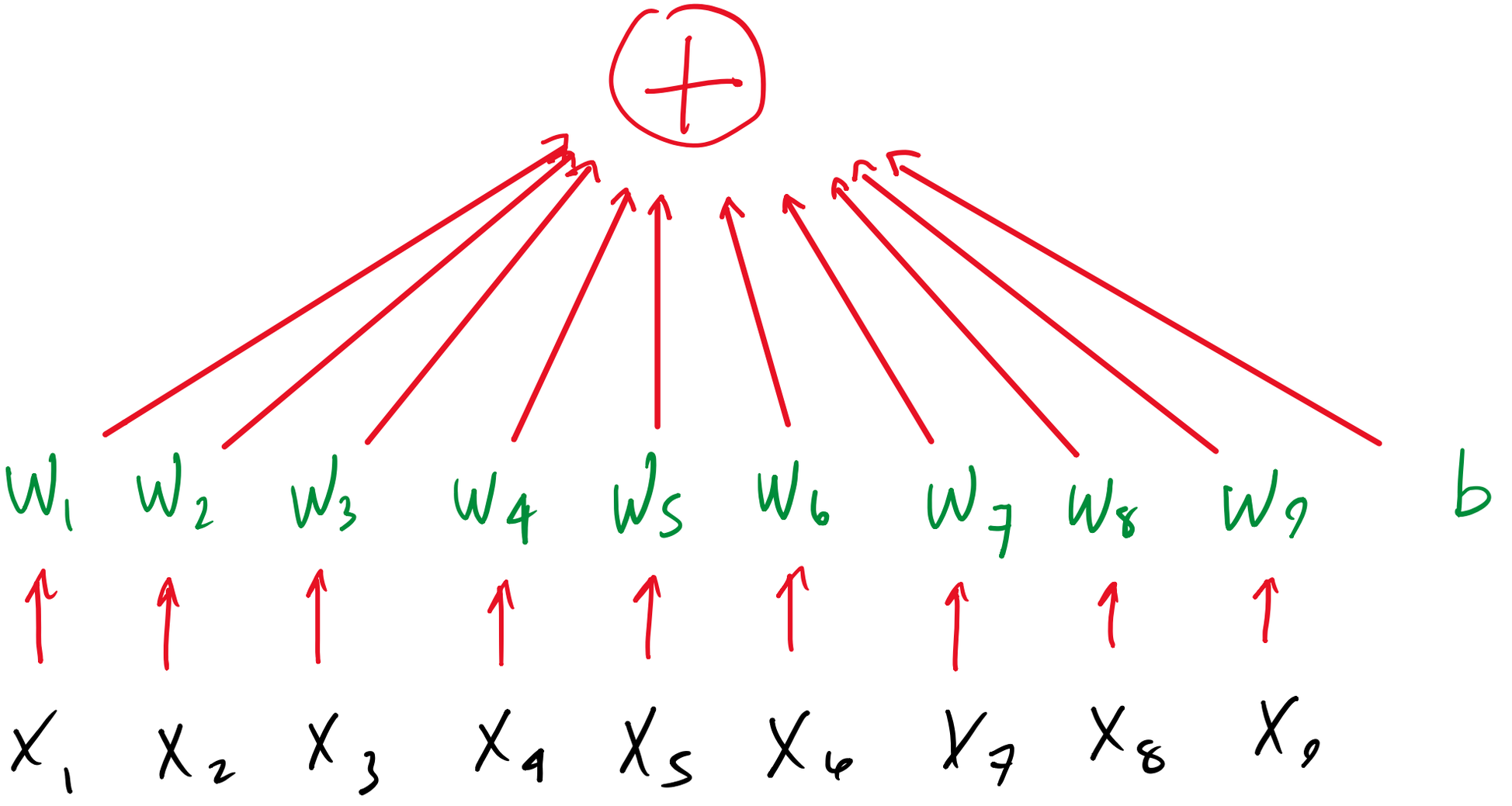


```
0,151 150 147 155 148 133 111 140 170 174 182 154 153 164 173 178 185 185 189 187 186 193 194 185 183 186 180 173 166 161 147 133 172 151 114 161 161 146 131 104 95 132 1
2,231 212 156 164 174 138 161 173 182 200 106 38 39 74 138 161 164 179 190 201 210 216 220 224 222 218 216 213 217 220 220 218 217 212 174 160 162 160 139 135 137 131 94 5
4,24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 19 43 52 13 26 40 59 65 12 20 63 99 98 98 111 75 62 41 73 118 140 192 186 187 188 190 190 187 182 176 173 172 173 25 34 29 1
6,4 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84 115 127 137 142 151 156 155 149 153 152 157 160 162 159 145 121 83 58 48 38 21 17 7 5 25 27 24 25 1 0 0 0 0 0 0 0 0 0 0 6
2,55 55 55 55 55 54 60 68 54 85 151 163 170 179 181 185 188 188 191 196 189 194 198 197 195 194 190 193 195 184 175 172 161 159 158 159 147 136 137 136 146 120 86 93 114 1
4,20 17 19 21 25 38 42 42 46 54 56 62 63 66 82 108 118 130 139 134 132 126 113 97 126 148 157 161 155 154 154 164 189 204 194 168 180 188 214 214 214 216 208 220 205 187 1
3,77 78 79 79 78 75 60 55 47 48 58 73 77 79 57 50 37 44 56 70 80 82 87 91 86 80 73 66 54 57 68 69 68 68 49 46 75 71 69 70 70 72 72 71 72 74 77 76 83 84 82 81 81 69 60 60 4
3,85 84 90 121 101 102 133 153 153 169 177 189 195 199 205 207 209 216 221 225 221 220 218 222 223 217 220 217 211 196 188 173 170 133 117 131 121 88 73 73 50 27 34 32 34
2,255 254 255 254 254 179 122 107 95 124 149 150 169 178 179 179 181 181 184 190 191 191 193 190 190 195 194 192 193 196 193 192 188 182 173 162 152 144 129 116 113 106 18
0,30 24 21 23 25 25 49 67 84 103 120 125 130 139 140 139 148 171 178 175 176 174 180 180 178 178 182 185 183 186 186 178 180 172 175 171 155 152 141 136 132 137 131 96 46
6,39 75 78 58 58 45 49 48 103 156 81 45 41 38 49 56 60 49 32 31 28 52 83 81 78 75 62 31 18 19 19 20 17 20 16 15 12 10 11 10 23 36 65 59 9 3 5 7 93 69 86 90 84 75 51 129 13
6,219 213 206 202 209 217 216 215 219 218 223 230 227 227 233 235 234 236 237 238 234 226 219 212 208 201 190 183 176 161 74 15 24 22 22 22 21 19 19 20 20 23 7 89 255 252
6,148 144 130 129 119 122 129 131 139 153 140 128 139 144 146 143 132 133 134 130 140 142 150 152 150 134 128 149 142 138 156 155 140 136 143 143 139 144 160 170 154 181 1
3,4 2 13 41 56 62 67 87 95 62 65 70 80 107 127 149 153 150 165 168 177 187 176 167 152 128 130 149 149 146 130 139 139 143 134 105 78 56 36 50 69 82 64 35 10 11 13 105 4 4
5,107 107 109 109 109 109 110 101 123 140 144 144 149 153 160 161 161 167 168 169 172 172 173 175 176 171 170 166 165 162 162 157 150 149 145 140 136 132 128 111 98 111 10
3,14 14 18 28 27 22 21 30 42 61 77 86 88 95 100 99 101 99 98 99 99 96 101 102 96 95 94 88 78 72 65 55 40 25 20 20 42 64 74 129 133 125 144 151 153 154 154 155 16 14 24 28
2,255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 2
6,134 124 167 180 197 194 203 210 204 203 209 204 206 211 211 216 219 224 228 230 230 226 222 220 217 217 210 207 213 210 199 191 190 188 177 172 148 142 90 46 81 132 116
4,219 192 179 148 208 254 192 98 121 103 145 185 83 58 114 227 225 220 203 202 168 154 157 164 182 211 164 94 122 155 176 238 240 242 192 87 43 39 60 85 34 57 82 130 113 1
4,1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 7 12 23 45 38 35 14 43 27 31 24 18 20 29 18 6 2 4 2 0 1 1 1 1 0 5 13 16 16 16 15 14 1 1 1 1 1 1 1 1 1 1 1 1 0 0 3 7 9 27 44 43 42 26 28 6
2,174 51 37 37 38 41 22 25 22 24 35 51 70 83 98 113 119 127 136 149 149 141 125 107 77 50 30 21 9 38 96 79 72 87 60 23 25 43 29 24 33 51 36 33 26 20 136 255 107 46 43 34 4
0,123 125 124 142 209 226 234 236 231 232 235 223 211 196 184 181 182 186 185 193 208 211 208 201 196 192 191 192 194 201 207 216 225 225 223 220 219 220 172 124 124 122 1
0,8 9 14 21 26 32 37 46 52 62 72 70 71 73 76 83 98 92 80 90 110 148 158 149 166 172 166 166 164 179 196 197 189 185 194 189 174 173 177 179 186 197 192 191 185 176 171 146
3,252 250 246 229 182 140 98 72 53 44 67 95 95 89 89 90 90 93 94 89 88 88 83 82 81 73 74 69 69 64 58 49 46 49 35 32 27 25 24 21 13 9 11 15 15 15 21 24 254 246 221 176 156
3,224 227 219 217 215 210 187 177 189 200 206 212 210 208 204 207 206 207 205 203 203 206 206 201 204 199 196 189 187 187 186 185 184 186 183 182 182 173 161 154 176 202 2
5,162 200 187 180 197 198 196 192 176 152 136 114 109 117 124 133 162 182 195 200 202 203 204 200 201 202 200 200 200 200 198 195 192 189 185 183 168 152 130 114 106 97 96
0,236 230 225 226 228 209 199 193 196 211 199 198 194 199 214 209 202 207 179 156 154 152 157 156 151 144 136 138 140 148 183 209 192 177 158 179 232 255 254 255 255 255 2
3,210 210 210 210 211 207 147 103 68 60 47 70 124 118 119 123 124 131 136 138 146 144 147 152 157 159 162 165 165 164 163 161 159 159 159 162 159 159 156 151 143 138 131 9
5,50 44 74 141 187 187 169 113 80 128 181 172 76 62 37 41 40 37 55 44 37 36 48 45 40 48 82 82 78 69 56 60 50 42 52 69 125 154 128 68 57 46 50 51 46 40 37 38 48 76 114 116
3,234 233 228 231 234 233 236 230 236 196 112 85 100 114 129 133 132 132 147 164 180 156 161 156 151 150 137 117 139 157 138 145 137 132 114 94 80 70 65 88 88 144 228 227
6,244 214 198 187 105 89 47 38 40 28 26 29 42 39 33 48 35 22 28 31 42 46 42 58 61 54 42 46 54 43 38 49 50 86 100 101 93 65 69 71 61 114 161 211 237 242 248 250 222 207 198
3,88 116 108 66 61 52 56 74 80 70 55 57 61 68 57 44 62 69 65 61 60 61 68 55 51 65 66 71 60 79 95 94 86 113 112 99 113 108 116 118 88 72 68 25 46 135 102 99 125 123 68 64 7
2,59 63 70 67 67 72 69 80 95 105 125 141 149 155 164 168 168 174 180 181 184 187 190 194 194 196 200 201 201 200 200 196 193 191 186 181 179 168 153 197 244 238 240 240 24
3,66 63 76 65 49 44 52 43 47 95 121 51 41 49 145 231 230 231 233 235 235 235 236 236 236 235 234 235 244 254 254 254 254 254 254 254 254 254 254 254 254 254 254 254 254 25
6,81 106 91 100 90 85 108 130 147 84 46 81 156 206 222 229 233 235 236 238 240 240 241 240 240 240 240 239 239 238 236 231 226 221 213 198 166 110 43 27 15 13 28 52 65 71
6,185 179 183 191 204 150 35 27 25 59 114 134 145 145 145 145 146 149 154 157 158 159 158 160 160 161 162 162 159 159 159 155 148 145 131 90 35 15 19 118 187 180 182 180 1
6,40 46 42 43 34 81 150 187 193 196 200 202 207 205 208 208 209 212 218 207 197 188 179 181 187 187 182 175 172 166 166 156 152 151 136 126 124 135 144 142 121 171 239 144
0,255 82 0 3 0 0 0 0 0 0 3 0 16 17 3 60 29 0 1 25 7 2 1 1 0 1 2 0 21 35 121 225 4 3 5 3 0 0 1 2 5 3 45 151 161 103 103 191 213 11 1 0 0 0 0 0 1 4 0 65 166 156 92 104 46 16
5,68 69 67 60 55 50 45 47 50 48 43 41 45 48 48 45 44 44 40 38 34 36 43 46 65 74 75 84 95 102 112 99 67 53 49 43 82 191 215 218 218 217 218 216 214 213 209 210 69 72 67 56
0,74 66 82 82 85 68 79 103 113 112 116 136 102 62 65 75 96 121 142 159 179 190 199 199 202 205 213 216 215 214 212 210 206 201 190 164 128 114 87 54 47 55 56 50 65 73 80 9
3,63 58 60 59 55 74 66 48 55 72 88 99 96 98 101 105 120 141 149 158 135 130 137 142 142 146 152 157 158 156 153 155 151 144 149 147 139 121 105 97 92 94 85 59 55 77 77 71
4,157 160 161 162 157 159 160 161 162 165 167 166 166 168 168 168 171 175 174 173 171 168 165 161 156 153 148 134 134 132 122 116 76 45 38 21 29 33 26 29 27 36 28 12 12 16
0,206 208 209 212 214 216 216 219 221 222 222 222 224 225 225 225 226 227 227 227 232 216 178 115 82 72 78 64 52 60 45 51 84 81 68 83 96 111 109 131 184 227 228 221 221 21
```

$$h(x)$$



```
1     1,  182,  199,   90,  100,  200,  247,  179,  199,    4
2     0,  189,   77,   19,   53,  196,  247,  100,  226,  250
3     1,  254,  188,  173,   48,   81,   83,   52,  166,   58
4     1,  244,  227,  139,   12,   79,  207,   15,   49,   53
5     1,  214,  189,  186,  102,   68,  232,  240,   97,    7
6     1,  123,  186,   31,   10,  222,  203,  113,  149,   23
7     1,  137,  249,  190,  153,   96,  160,  186,   55,  242
8     0,   97,  238,   22,  210,  253,   38,  211,  175,  253
9     0,  130,  176,  155,   80,   73,  212,  251,   96,  133
10    1,   74,  159,    7,   16,   38,    1,   82,  108,   34
11    0,  205,  126,  158,  166,   39,  204,  251,  239,  168
12    0,  182,  227,   79,  170,  204,  160,  220,  224,   61
13    0,  174,  241,   64,   11,  202,  252,  165,  190,  205
14    0,  202,   14,  154,   72,  107,  140,  149,  234,   18
15    0,  112,  179,   31,  215,  104,   13,  233,  150,  115
16    0,  104,   59,   64,  217,   28,  146,  244,  238,   95
17    0,   83,   60,  172,  150,   15,  196,    3,  244,  250
18    1,  101,  111,  165,   64,   19,   61,   94,   10,  238
19    1,   23,  226,  109,  110,   94,  221,    1,   82,  147
20    0,   21,   20,  183,  114,  125,  148,  222,  111,  162
21    0,  144,  101,   98,  138,    2,   33,   70,  233,   59
22    0,  114,  117,   84,  210,   67,   91,    1,  240,  230
23    0,  203,   50,  216,   86,   74,   75,  125,  219,  233
24    0,   27,  217,  100,  102,  104,   30,  111,  234,   65
25    0,   29,  120,   62,   42,  149,  223,  202,   37,  132
26    1,  206,   46,  182,  248,  151,   15,  202,   56,    4
27    0,  134,  206,  224,    4,   19,  180,  201,  173,  220
28    1,   95,  195,  159,   26,   11,   41,  147,  143,   71
29    1,   47,  187,  200,   70,  187,  165,  208,   17,  140
30    1,   10,  232,  107,   36,  172,  189,  102,  111,    6
31    0,  223,   60,   31,  107,  109,  232,  238,  131,   57
32    0,   61,   36,   78,   41,  137,  213,  202,  106,  237
33    1,  188,  220,  178,  222,  109,   12,  113,  195,   58
34    1,   52,  187,  219,  149,   64,   28,  219,  196,   28
35    1,  207,  132,   99,  114,   23,  238,   19,  223,  182
```

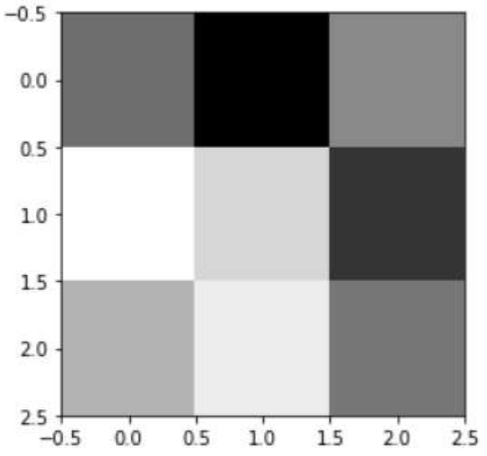

+
w1 w2 w3 w4 w5 w6 w7 w8 w9 b
x1 x2 x3 x4 x5 x6 x7 x8 x9

$$X = [x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9]$$

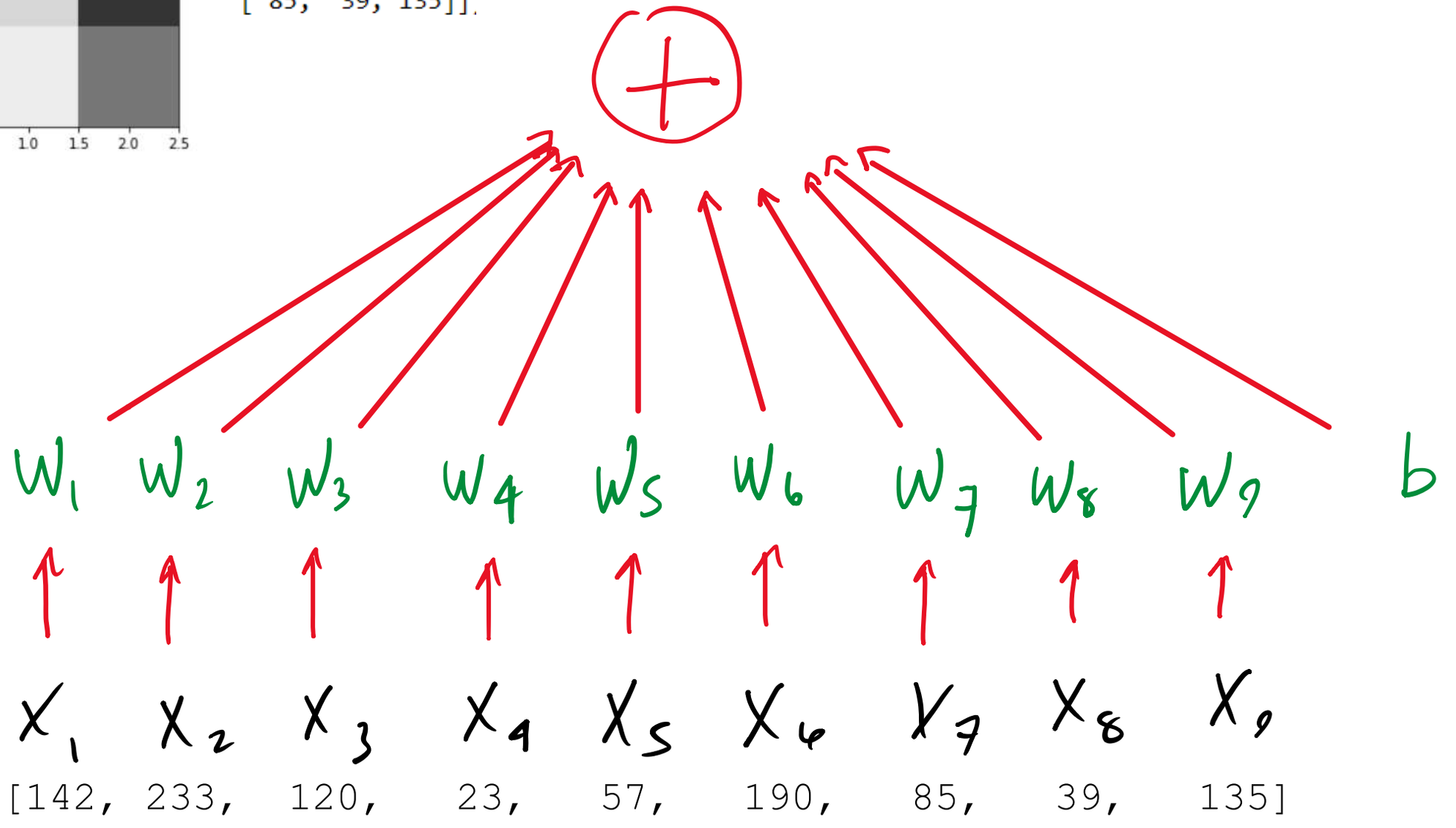
$$w = [w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9]$$

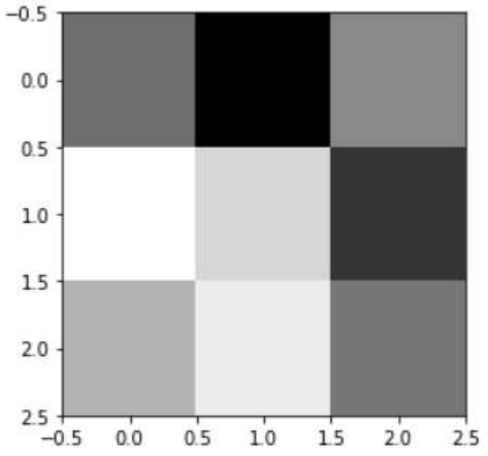
$$w^T x + b$$

$$w^T x = \|w\| \cdot \|x\| \cdot \cos\theta$$

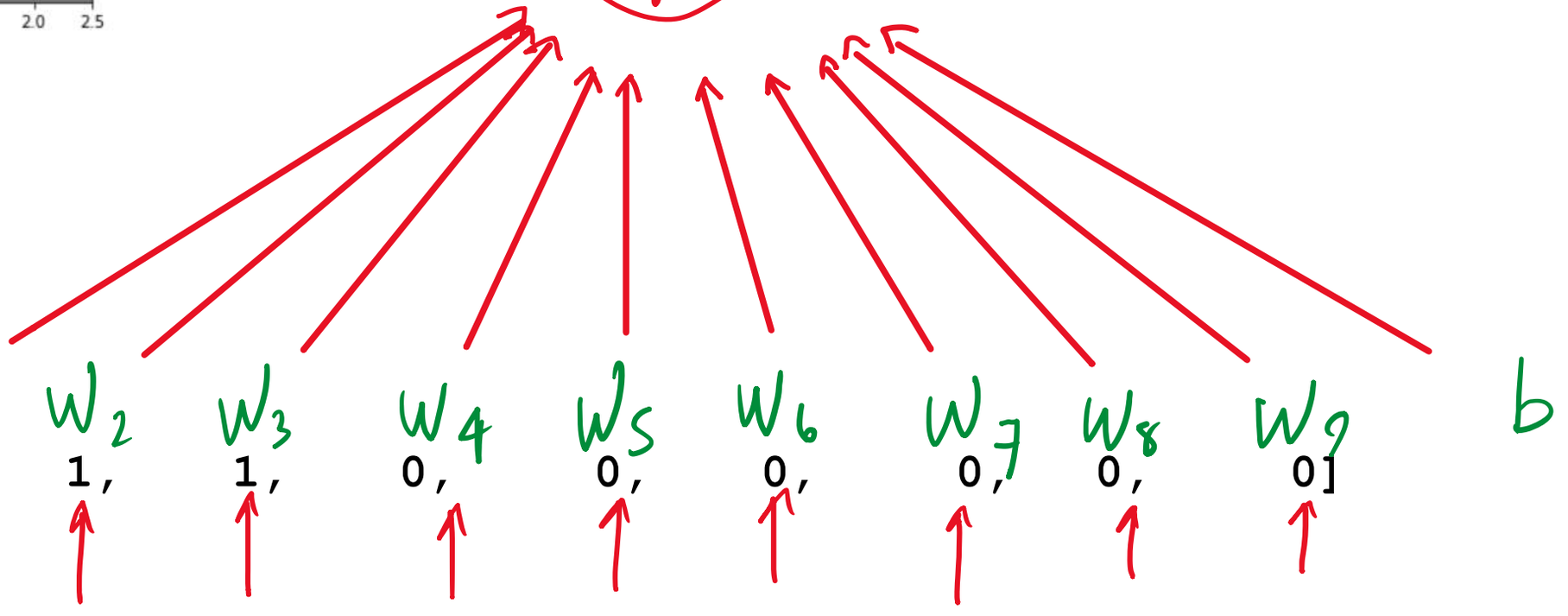


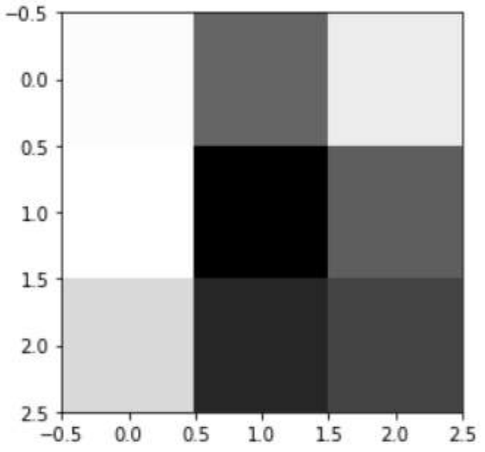

```
[[142, 233, 120],
 [ 23,  57, 190],
 [ 85,  39, 135]]
```



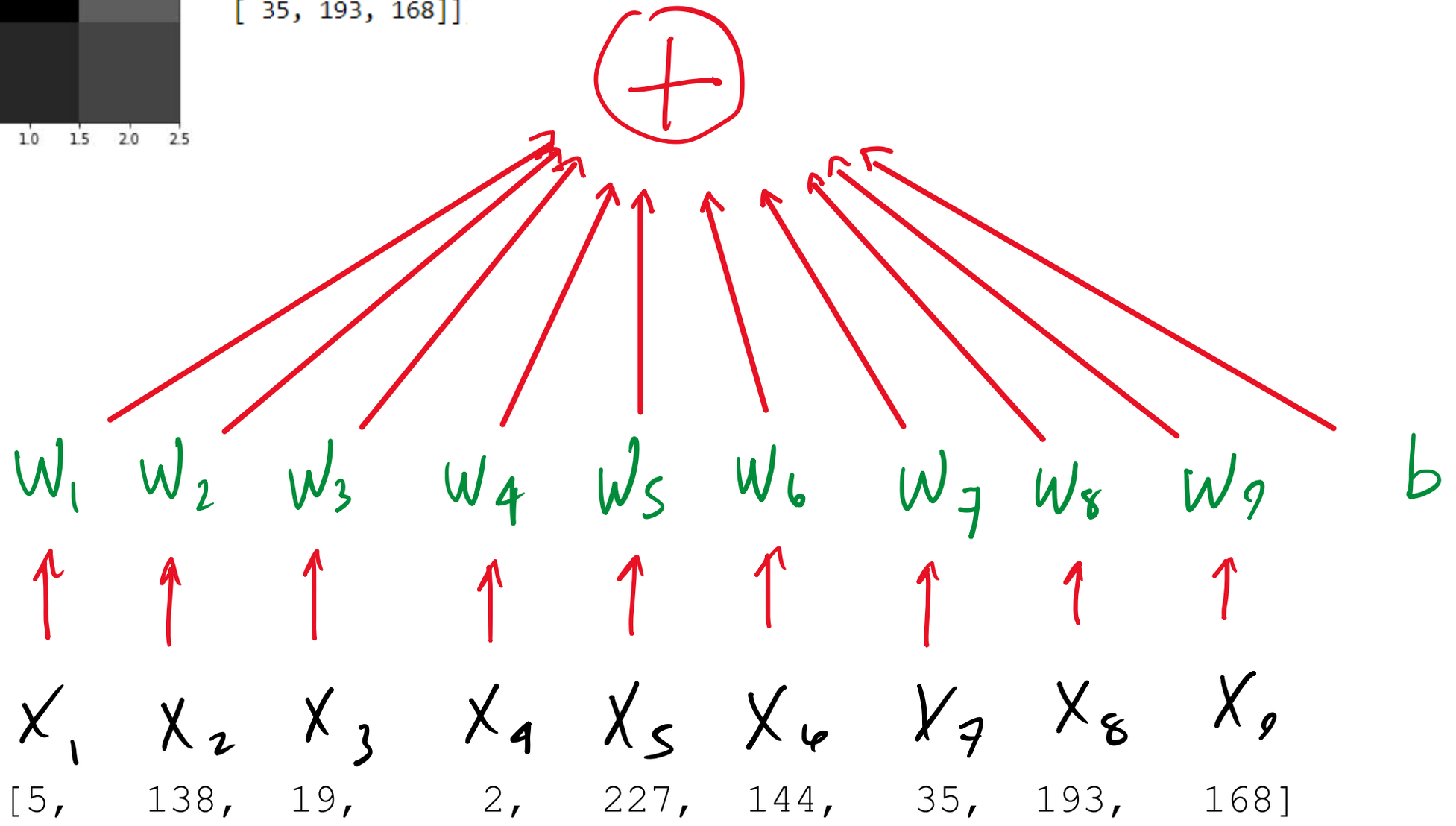


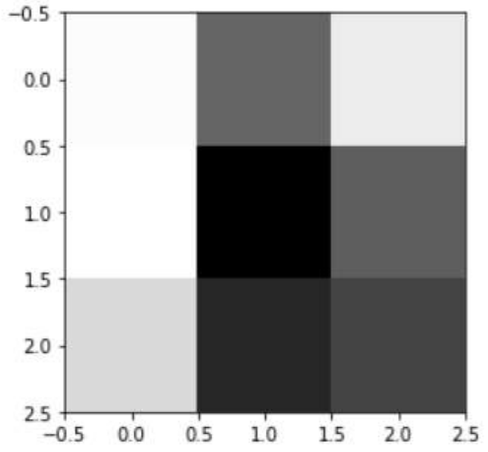

```
[[142, 233, 120],
 [ 23,  57, 190],
 [ 85,  39, 135]]
```



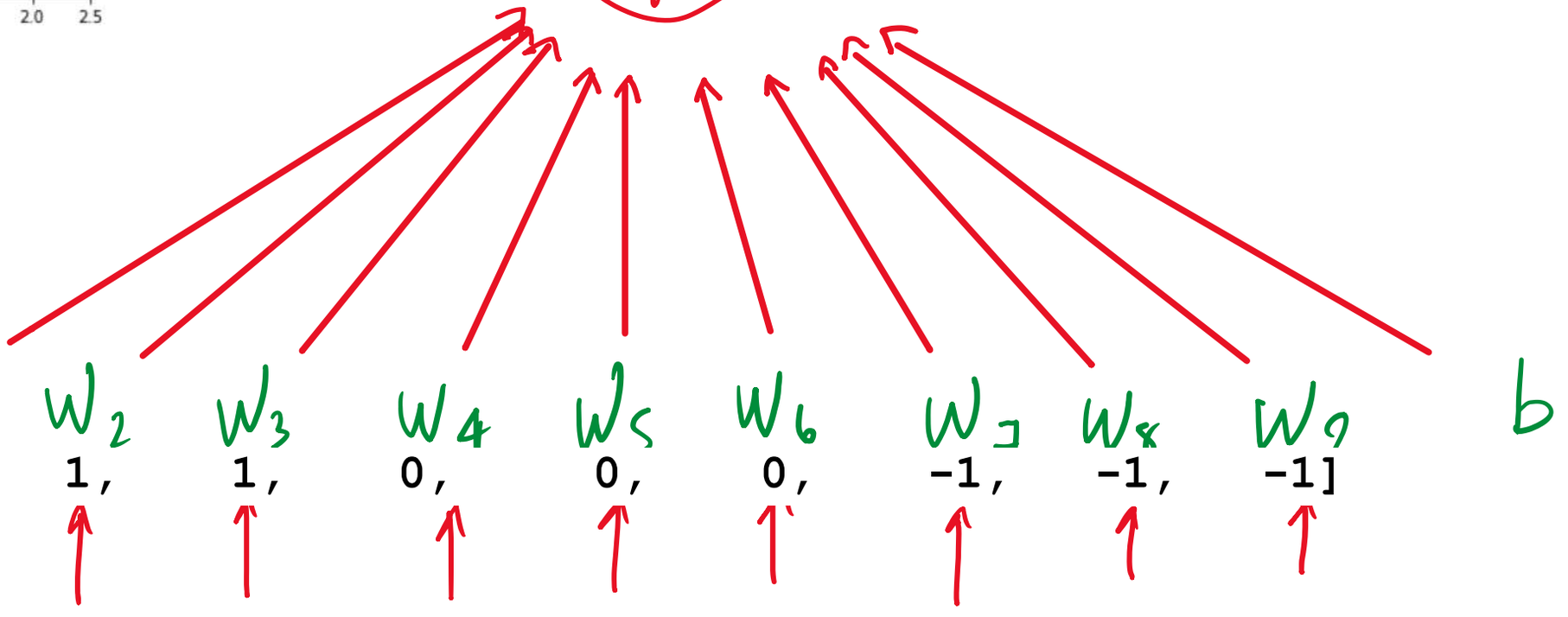



```
[[  5, 138,  19],
 [  2, 227, 144],
 [ 35, 193, 168]]
```






```
[[  5, 138,  19],
 [  2, 227, 144],
 [ 35, 193, 168]]
```

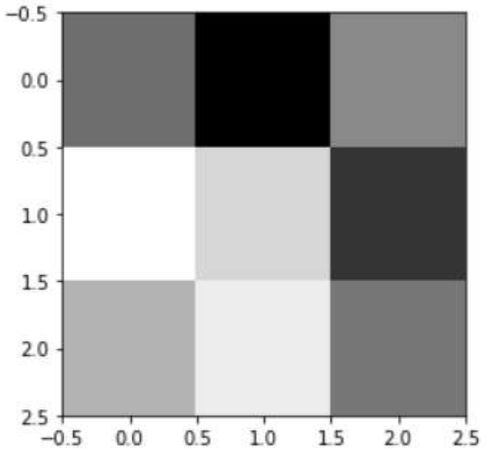


w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 w_9 b

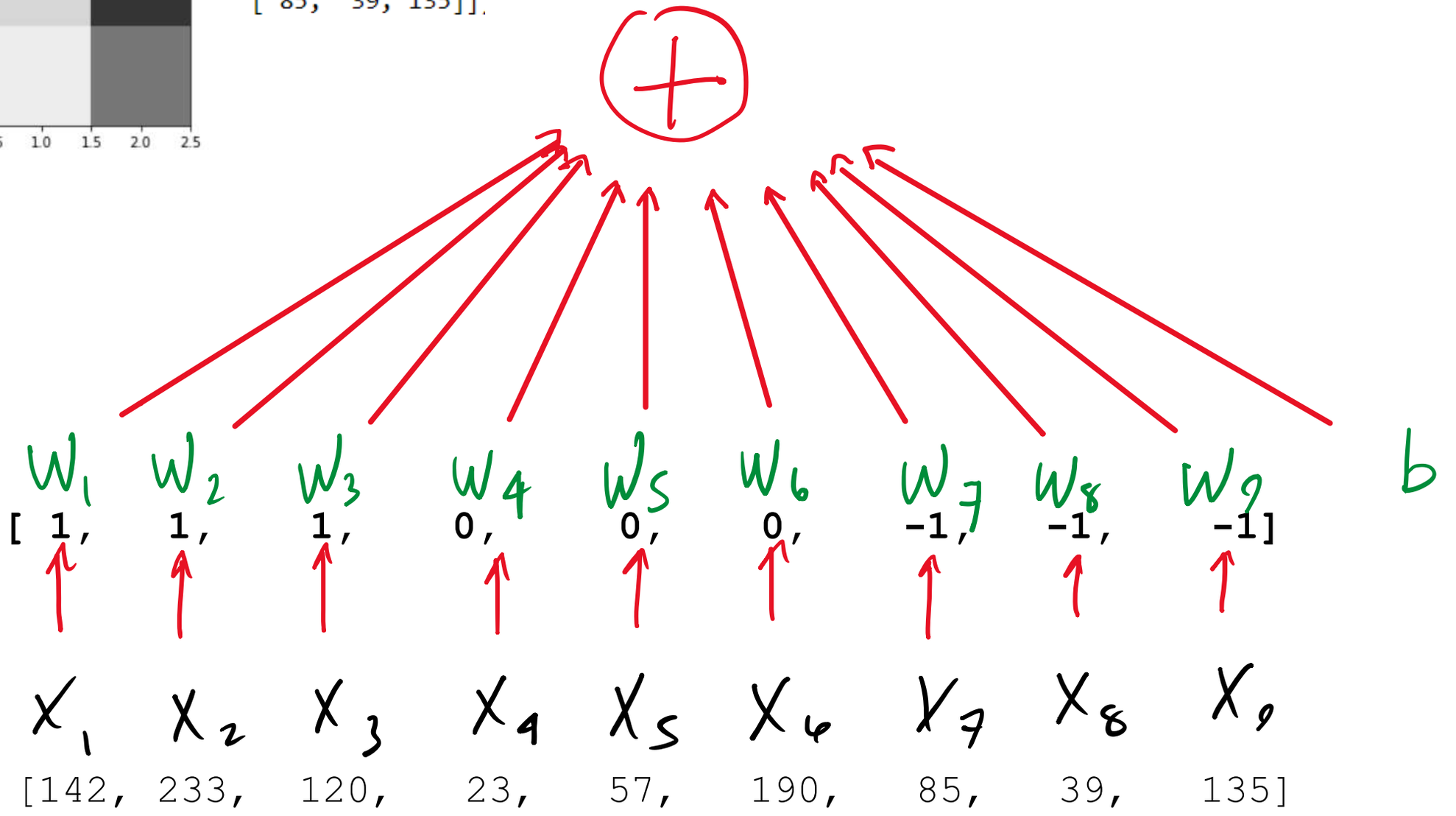
[1, 1, 1, 0, 0, 0, -1, -1, -1]

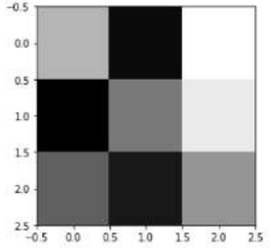
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9

[5, 138, 19, 2, 227, 144, 35, 193, 168]

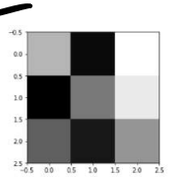



```
[[142, 233, 120],
 [ 23,  57, 190],
 [ 85,  39, 135]]
```



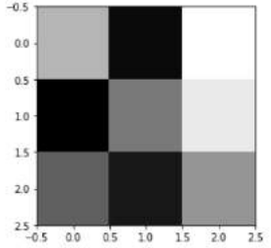


w^T

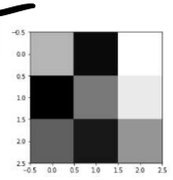


+ = top

− = bottom



w^T



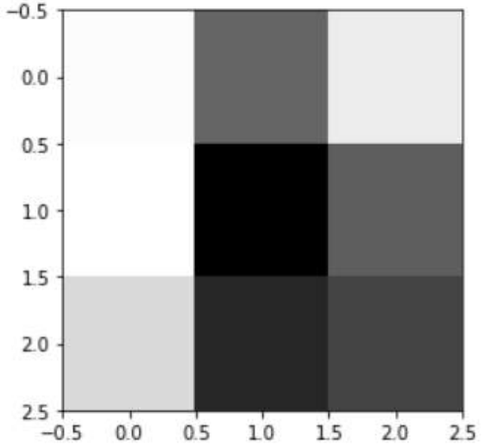
+ = top

− = bottom

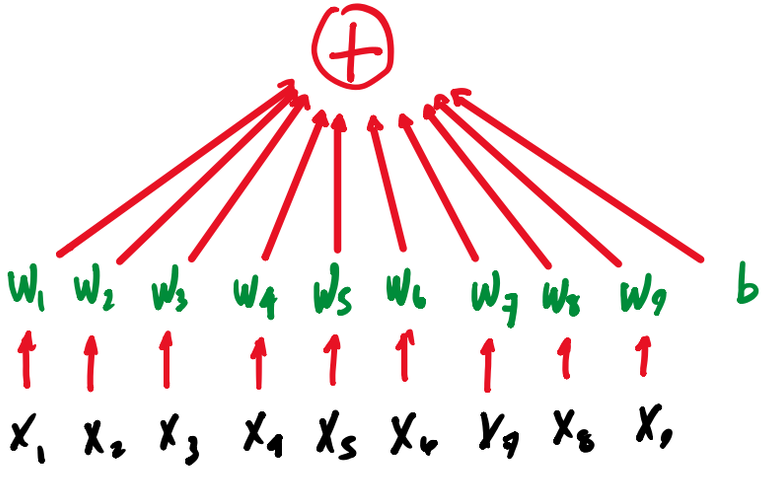
model

prediction

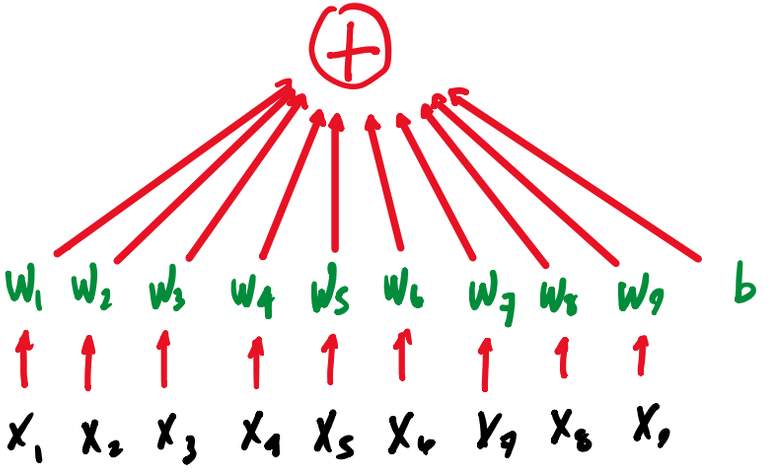
```
[[  5, 138,  19],
 [  2, 227, 144],
 [ 35, 193, 168]]
```



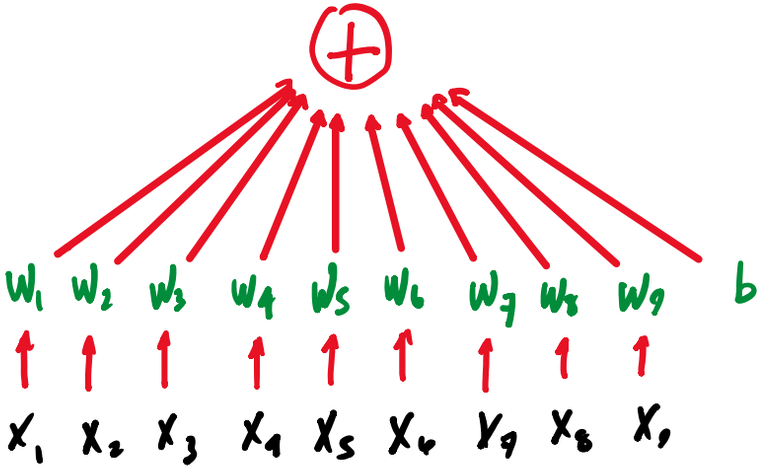
top / middle / bottom ?



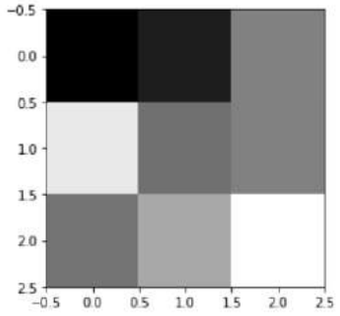
top



middle



bottom



```
[[200, 179, 107],
 [ 34, 120, 108],
 [118,  80,  18]]
```

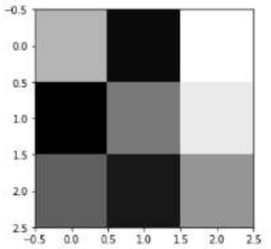
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 200 \\ 179 \\ 107 \\ 34 \\ 120 \\ 108 \\ 118 \\ 80 \\ 18 \end{bmatrix} = \begin{bmatrix} 486 \\ 262 \\ 216 \end{bmatrix}$$

$$W \qquad X$$

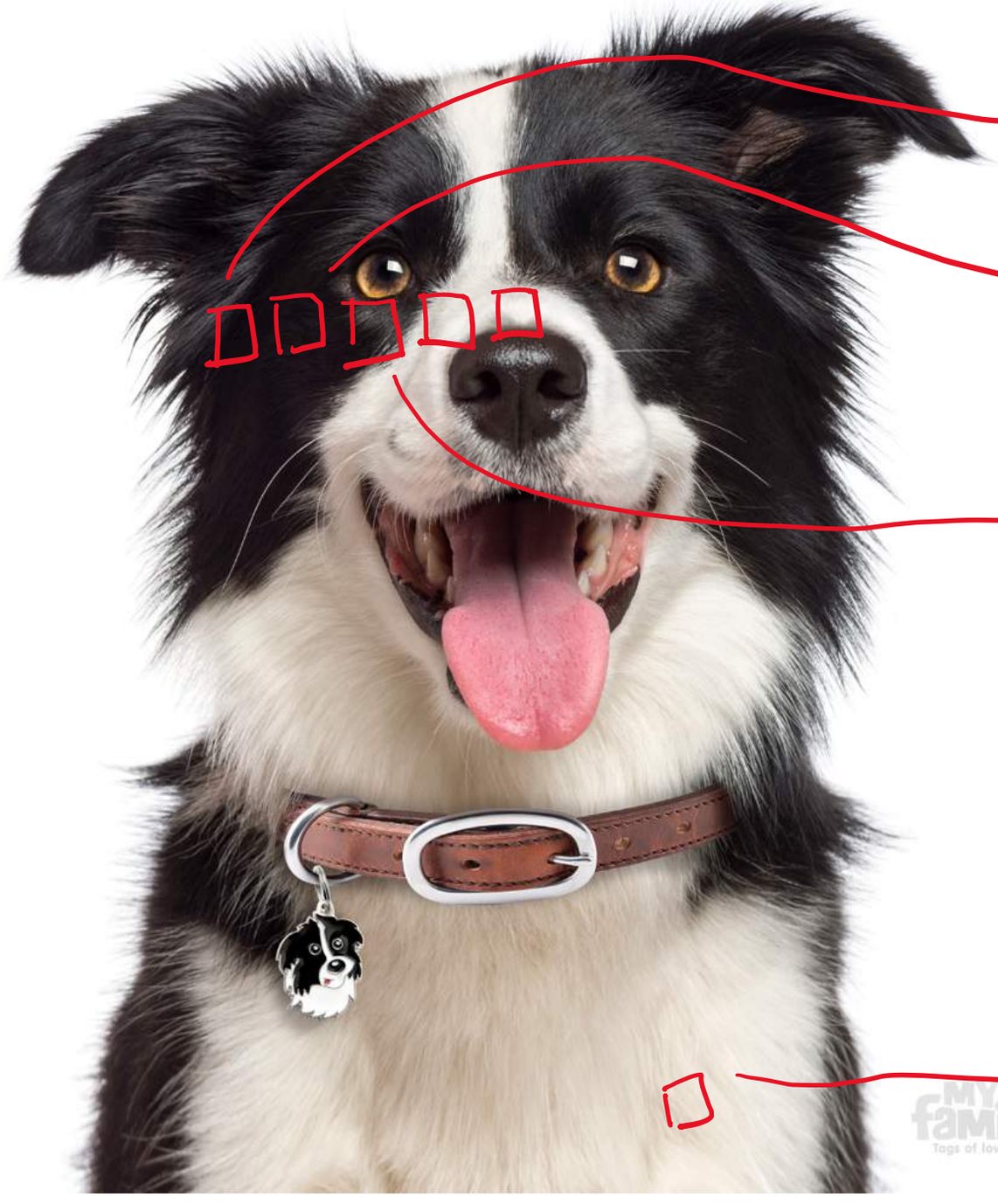
$$\varphi(\quad) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

top / middle / botton

$$\varphi\left(W^{T}x + b\right)$$



→ dog | cat



$$\varphi(w_1^T x + b_1) \rightarrow$$

$$\varphi(w_2^T x + b_2) \rightarrow$$

$$\varphi(w_3^T x + b_3) \rightarrow$$

$$\vdots$$

$$\varphi(w_n^T x + b_n) \rightarrow$$

$\varphi(w_1^T x + b_1) \rightarrow$

$\varphi(w_2^T x + b_2) \rightarrow$

$\varphi(w_3^T x + b_3) \rightarrow$

$\vdots$

$\varphi(w_n^T x + b_n) \rightarrow$

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix} \longrightarrow$$

do it again!

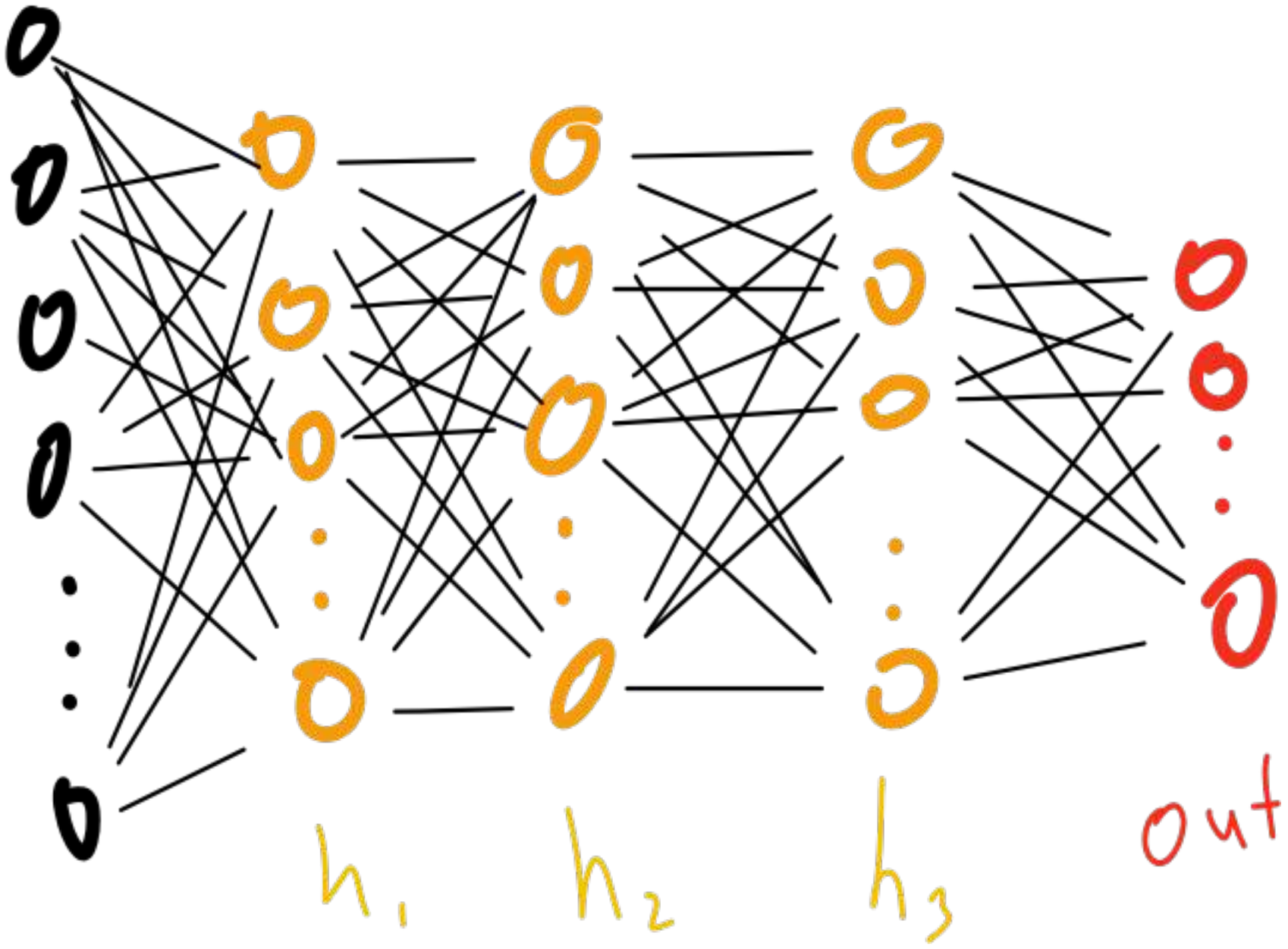
$\varphi(w_1^T x + b_1) \rightarrow$

$\varphi(w_2^T x + b_2) \rightarrow$

$\varphi(w_3^T x + b_3) \rightarrow$

$\vdots$

$\varphi(w_n^T x + b_n) \rightarrow$



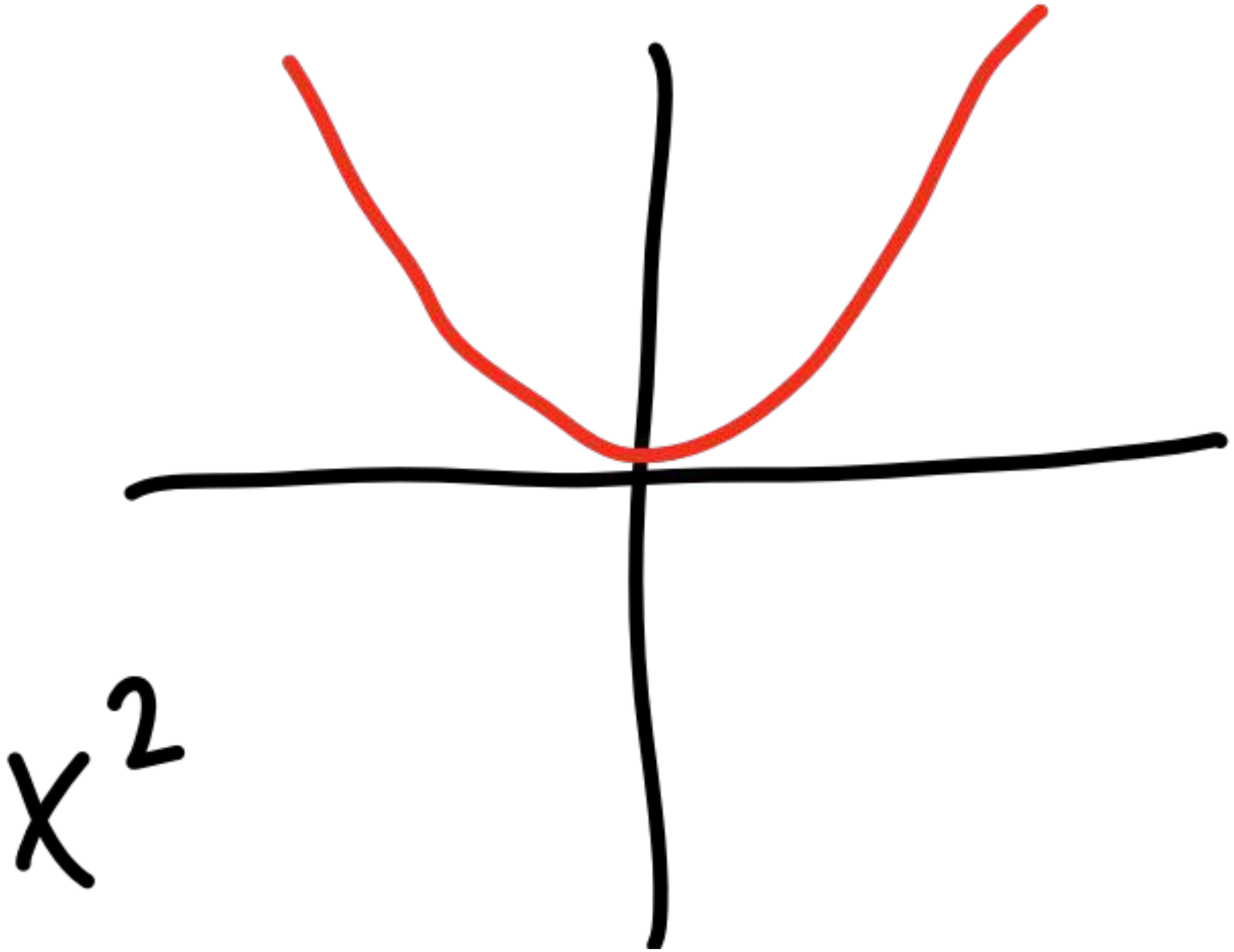
how do we get all the W's + b's ??

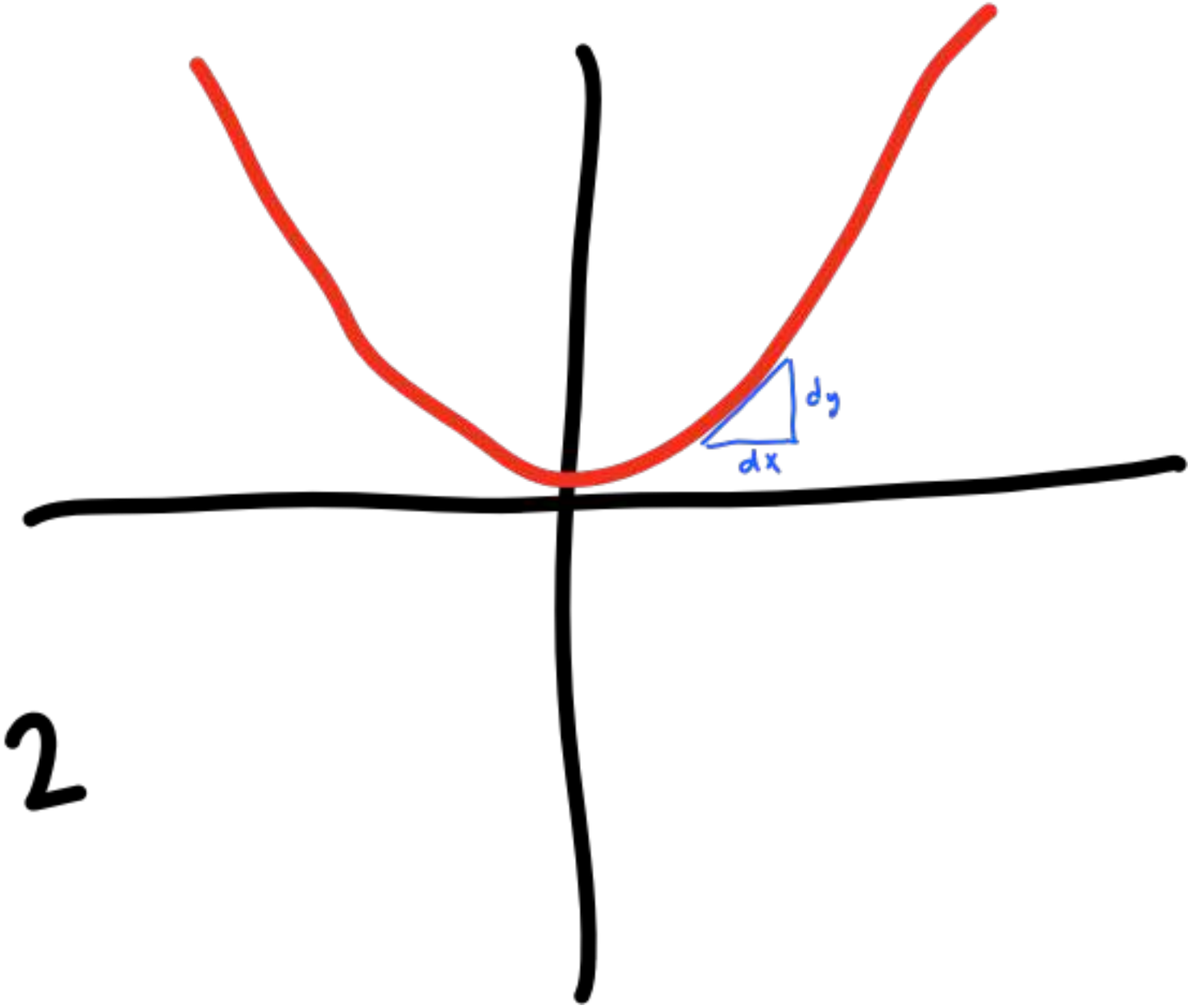
we want to minimize mistakes

$$J(W_s, b_s) =$$

$$\frac{1}{n} \sum_{n} \| h(x) - y \|^2$$

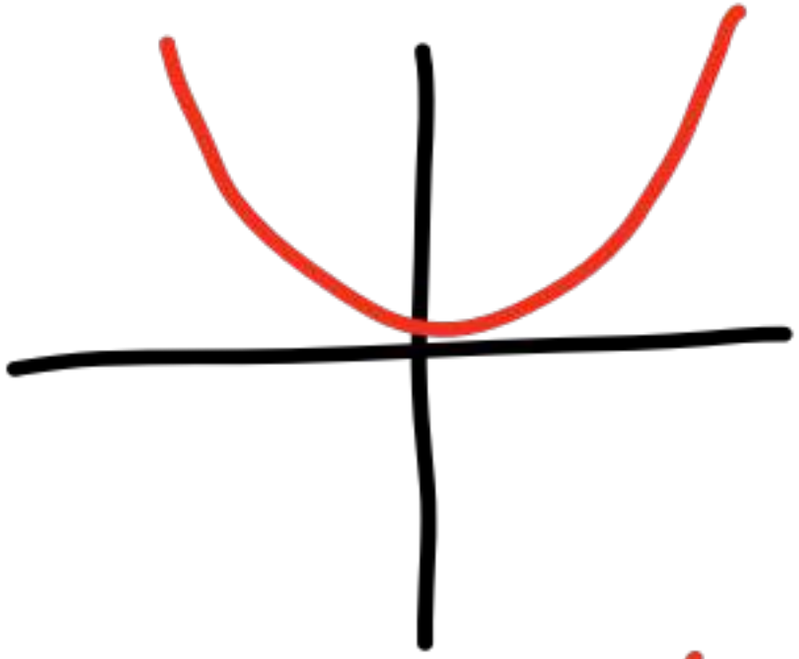
$W's, b's$


x^2

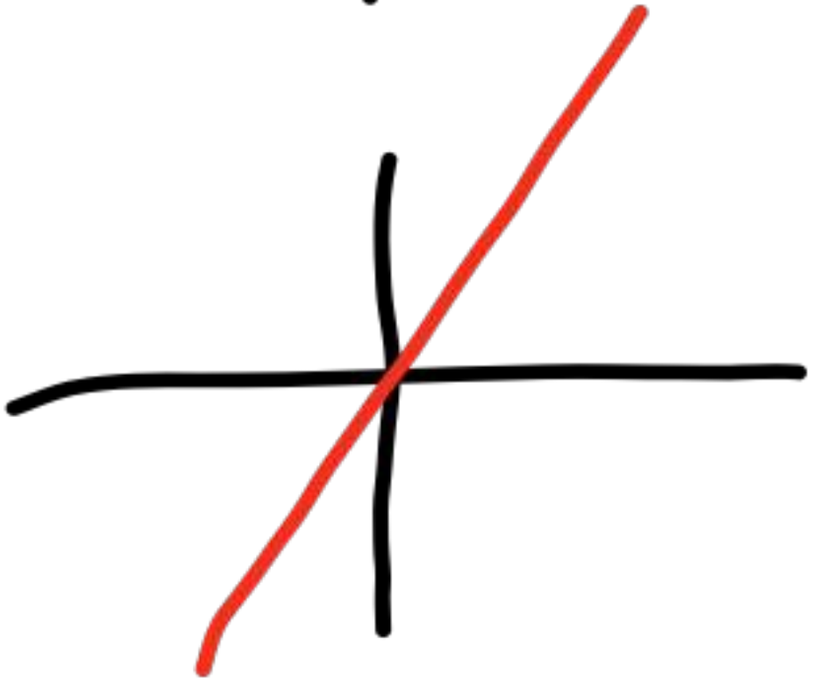


x^2

$$x^2$$



$$\frac{d}{dx} x^2 = 2x$$



learning rate

$$W_1 := W_0 - \alpha \frac{d}{dw} J(w)$$

updated W

current W

derivative of cost function

learning rate

$$W_1 := W_0 - \alpha \nabla J(W_0)$$

updated W

current W

gradient of cost function

so far…

- x – incoming images
- y – classification answer
- model function –
- loss / cost function – mean squared error
- optimization method – gradient descent

Demo: Deep Learning in Action

find out more!

- Learn more about
  - Cognitive Services – https://aka.ms/cogsvcs
  - Azure ML – https://aka.ms/azuremlbuild
  - Visual Studio Tools for AI - https://aka.ms/vstoolsforai
  - Deep Learning VMs - https://aka.ms/dlvmbuild
  - Batch AI – https://aka.ms/batchaibuild
  - ML.Net - https://docs.microsoft.com/en-us/dotnet/machine-learning/
- azure.com/ai

Demo: Azure Functions

Thank you!
#azuredevtour

@azureadvocates

aka.ms/devtourcode