Non-vectorised:

$u = Av$

$u_i = \sum_j A_{ij} v_j$

```
u = np.zeros((n,1))
for i ...
    for j ...
        u[i] += A[i][j]*v[j]
```

Vectorised:

```
u = np.dot(A,v)
```

Vectors and matrix valued functions:

E.g.: apply exp to every element of a vector/matrix:

$$v = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix} \qquad u = \begin{bmatrix} e^{v_1} \\ \vdots \\ e^{v_n} \end{bmatrix}$$

NV:

```
u = np.zeros((n,1))
for i in range(n):
    u[i] = math.exp(v[i])
```

V:

```
import numpy as np
u = np.exp(v)
```

np.log(v)
np.fabs(v)
np.maximum(
.
;

Log. reg. derivatives:

~~$dw_1 = 0$, $dw_2 = 0$~~ ... `dw = np.zeros((n_x,1))`

~~$dw_1 += x^{(i)} dz^{(i)}$~~
~~$dw_2 += x_2^{(i)} dz^{(i)}$~~

`dw += x^(i) dz^(i)`

~~$dw_1 = dw_1/m$, $dw_2 = dw_2/m$~~

`dw /= m`