< Vectorized implementations of neural network >

[For Loop]

```
x = np.array([200, 17])

W = np.array([[1, -3, 5],
              [-2, 4, 6]])

b = np.array([-1, 1, 2])

def dense(a_in, W, b):
    units = W.shape[1]
    a_out = np.zeros(units)
    for j in range(units):
        w = W[:, j]
        z = np.dot(w, x) + b[j]
        a[j] = g(z)
    return a_out
```

⇒ [1, 0, 1]

[Vectorization]

```
X = np.array([[200, 17]])

W = np.array([[1, -3, 5],
              [-2, 4, -6]])

B = np.array([[-1, 1, 2]])
```

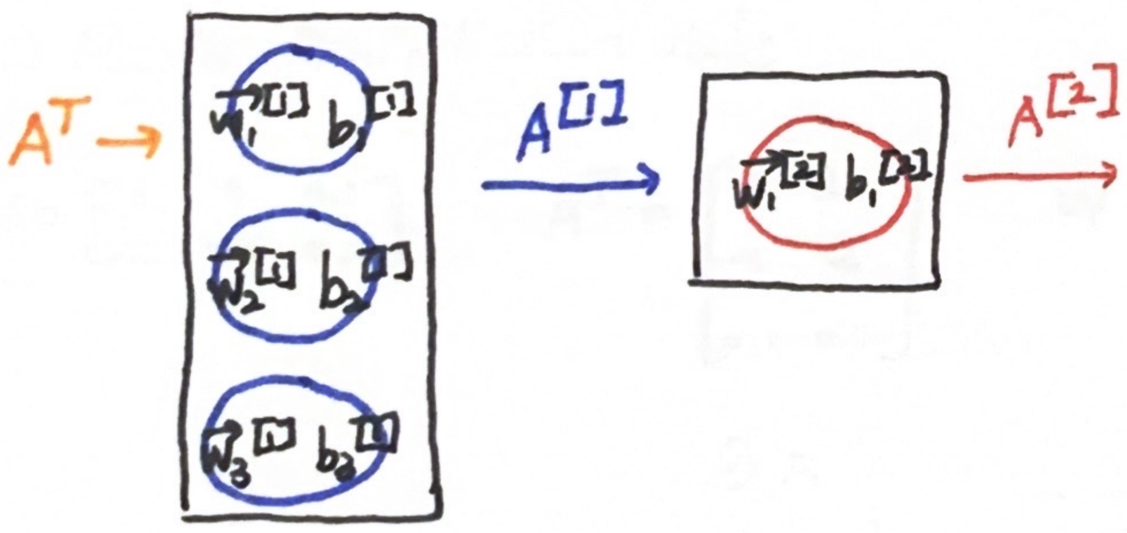
all 2D arrays

```
def dense(A_in, W, B):
    Z = np.matmul(A_in, W) + B
    A_out = g(Z)
    return A_out
```

matrix multiplication

⇒ [[1, 0, 1]]

* Matrix multiplication



$A = \begin{bmatrix} 200 \\ 17 \end{bmatrix}$ ← Transpose $A^T = \begin{bmatrix} 200 & 17 \end{bmatrix}$ 1×2

$W = \begin{bmatrix} 1 & -3 & 5 \\ -2 & 4 & -6 \end{bmatrix}$ 2×3

$\vec{b} = \begin{bmatrix} -1 & 1 & 2 \end{bmatrix}$ 1×3

$Z = A^T W + B = \begin{bmatrix} 165 & -531 & 900 \end{bmatrix}$

$A_{out} = g(Z) = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}$

apply sigmoid function on matrix Z