多層級RNN

程式碼：

def forward(self, X):

T = X.shape[0]

self.h = [np.zeros((T, self.n\_hidden\_units)) for \_ in range(self.n\_hidden\_layers)]

self.y = np.zeros((T, self.n\_output))

for t in range(T):

self.h[0][t] = np.tanh(np.dot(self.U[0], X[t]) + np.dot(self.W[0], self.h[0][t-1]) + self.b[0])

for l in range(1, self.n\_hidden\_layers):

self.h[l][t] = np.tanh(np.dot(self.U[l], self.h[l-1][t]) + np.dot(self.W[l], self.h[l-1][t-1]) + self.b[l])

self.y[t] = np.dot(self.V, self.h[-1][t]) + self.c

return self.y