Backpropagation through the Singular Value Decomposition

Viraj Navkal

Singular Value Decomposition

Assume $A \in M_{m \times n}(F)$, with $F = \mathbb{R}$ or $\mathbb{C}$.

Then there is a decomposition $A = USV^*$, with

1. $U^*U = I$,
2. $V^*V = I$, and
3. S diagonal with nonnegative real entries.

Two possible choices for dimensions:

1. `full_matrices` false:
   U: $m \times min(m, n)$
   V: $n \times min(m, n)$
   S: $min(m, n) \times min(m, n)$
2. `full_matrices` true:
   U: $m \times m$
   V: $n \times n$
   S: $m \times n$

Backgpropagation through SVD in TensorFlow

Initial implementation had some restrictions:

1. $F = \mathbb{R}$, not $\mathbb{C}$
2. $|m - n| \leq 1$
3. `full_matrices` true

Backgpropagation through SVD in TensorFlow

Which restrictions can be removed?

1. $F = \mathbb{R}$, not $\mathbb{C}$

   no – indeterminacy when $F = \mathbb{C}$
2. $|m - n| \leq 1$

   no, at least not when `full_matrices` true (indeterminacy)
3. `full_matrices` true

   yes – in fact, gradient formula is known

I implemented backpropagation in TensorFlow when `full_matrices` is false, for arbitrary m and n.

Derivation of dS

$$\begin{aligned} A &= USV^* \\ \Rightarrow \qquad dA &= dUSV^* + UdSV^* + USdV^* \\ \Rightarrow \quad U^*dAV &= U^*dUS + dS + SdV^*V \end{aligned}$$

Now, U^*dU is skew-Hermitian:

$$\begin{aligned} U^*U &= I \\ \Rightarrow \qquad dU^*U + U^*dU &= 0 \\ \Rightarrow \quad (U^*dU)^* + U^*dU &= 0 \end{aligned}$$

Let $\circ$ denote the element-wise product. Then

$$\begin{aligned} I \circ (U^*dAV) &= I \circ (U^*dUS) + I \circ dS + I \circ (SdV^*V) \\ &= dS \end{aligned}$$

Therefore $\boxed{dS = I \circ (U^*dAV)}$.

One possible application

Nuclear norm cost penalty for regularization or low-rank compression

Closing Question

The gradient calculation fails wherever the SVD function is not continuous. Is it possible to choose a function that is continuous everywhere?