

न्यूरल नेटवर्क से ००० तक

००००००० वीडियो

एंजेल कार्पथी - चलो ००० बनाएं: शुरुआत से, कोड में, विस्तार से।

उमर जमील - ध्यान ही सब कुछ है (ट्रांसफॉर्मर) - मॉडल व्याख्या (गणित सहित), अनुमान और प्रशिक्षण

स्टैटवेस्ट विथ जोश स्टारमर - ट्रांसफॉर्मर न्यूरल नेटवर्क्स, ००००००० का आधार, स्पष्ट रूप से समझाया गया!!!

पास्कल पूपार्ट - ००४८०/६८० लेक्चर १९: अटेंशन और ट्रांसफॉर्मर नेटवर्क्स

ए.आई. हैकर - माइकल फाई - ट्रांसफॉर्मर्स न्यूरल नेटवर्क का सचित्र गाइड: एक चरण-दर-चरण व्याख्या

मैं कैसे सीखता हूँ

जब मैंने “०००००० ०००००००० ००० ०००० ००००००००” पुस्तक का आधा हिस्सा पढ़ लिया, तो मैंने हस्तलिखित अंकों को पहचानने के न्यूरल नेटवर्क उदाहरण को दोहराना शुरू किया। मैंने ०००००० पर एक रिपोजिटरी बनाई, ०००००://००००००.०००/०००००००/००००००-००००००००-०००-००००००-००००००००।

यही असली कठिन हिस्सा है। अगर कोई बिना किसी कोड को कॉपी किए इसे शुरू से लिख सकता है, तो वह इसे बहुत अच्छी तरह समझता है।

मेरे ००००००००० कोड में अभी भी `update_mini_batch` और `backprop` का कार्यान्वयन नहीं है। हालांकि, डेटा लोड करने, फीड फॉरवर्डिंग, और मूल्यांकन के चरण में चरों को ध्यान से देखकर, मैंने वेक्टर, आयाम, मैट्रिक्स, और ऑब्जेक्ट्स के आकार को बहुत बेहतर ढंग से समझ लिया है।

और मैंने ००० और ट्रांसफॉर्मर के कार्यान्वयन को सीखना शुरू किया। शब्द एम्बेडिंग और पोजिशनल एन्कोडिंग के माध्यम से, टेक्स्ट संख्याओं में बदल जाता है। फिर, मूल रूप से, यह हस्तलिखित अंकों को पहचानने वाले सरल न्यूरल नेटवर्क से कोई अंतर नहीं रखता।

०००००० ०००००००० का लेक्चर “०००’० ००००० ०००” बहुत अच्छा है। वह चीजों को अच्छे से समझाते हैं।

पहला कारण यह है कि यह वास्तव में शुरुआत से है। हम पहले यह देखते हैं कि टेक्स्ट कैसे जनरेट किया जाता है। यह थोड़ा अस्पष्ट और यादृच्छिक होता है। दूसरा कारण यह है कि ०००००० चीजों को बहुत सहज तरीके से समझा सकते हैं। ०००००० ने ००००००० प्रोजेक्ट को कई महीनों तक किया।

मुझे लेक्चर की गुणवत्ता का आकलन करने के लिए एक नया विचार आया है। क्या लेखक वास्तव में ये कोड लिख सकता है? मुझे क्यों समझ में नहीं आ रहा है और लेखक ने कौन सा विषय छोड़ दिया है? इन सुंदर डायग्राम और एनिमेशन के अलावा, उनकी कमियाँ और दोष क्या हैं?

मशीन लर्निंग के विषय पर वापस आते हैं। जैसा कि ०००००० ने उल्लेख किया है, ड्रॉपआउट, रेज़िडुअल कनेक्शन, सेल्फ-अटेंशन, मल्टी-हेड अटेंशन, और मास्कड अटेंशन।

ऊपर दिए गए और वीडियो देखकर, मैं थोड़ा समझने लगा।

साइन और कोस फ़ंक्शन के साथ पोजिशनल एन्कोडिंग द्वारा, हमें कुछ वेट मिलते हैं। वर्ड एम्बेडिंग द्वारा, हम शब्दों को संख्याओं में बदलते हैं।

$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{model}}) PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

(यह गणितीय समीकरण है, इसे अनुवादित नहीं किया जाता है।)

पिज़्ज़ा ओवन से निकला और इसका स्वाद अच्छा था।

इस वाक्य में, एल्गोरिदम कैसे जानता है कि यह पिज़्ज़ा या ओवन को संदर्भित करता है? हम वाक्य में हर शब्द के लिए समानताओं की गणना कैसे करते हैं?

हमें वज़नों का एक सेट चाहिए। यदि हम अनुवाद का कार्य करने के लिए ट्रांसफॉर्मर नेटवर्क का उपयोग करते हैं, तो हर बार जब हम एक वाक्य इनपुट करते हैं, यह दूसरी भाषा में संबंधित वाक्य आउटपुट कर सकता है।

यहां डॉट प्रोडक्ट के बारे में। हम यहां डॉट प्रोडक्ट का उपयोग करने का एक कारण यह है कि डॉट प्रोडक्ट वेक्टर में हर संख्या को ध्यान में रखेगा। अगर हम स्क्वेर्ड डॉट प्रोडक्ट का उपयोग करें तो क्या होगा? हम पहले संख्याओं का वर्ग निकालते हैं, फिर उन्हें डॉट प्रोडक्ट करते हैं। अगर हम कुछ उल्टा डॉट प्रोडक्ट करें तो क्या होगा?

मास्किंग के बारे में, हम मैट्रिक्स के आधे हिस्से के नंबरों को नेगेटिव इनफिनिटी में बदल देते हैं। और फिर हम सॉफ्टमैक्स का उपयोग करके मानों को 0 से 1 के बीच में लाते हैं। अगर हम बाएं-नीचे के नंबरों को नेगेटिव इनफिनिटी में बदल दें तो क्या होगा?

योजना

कोड और पेपर्स पढ़ना और वीडियो देखना जारी रखें। बस मज़े करो और अपनी जिज्ञासा का पालन करो।

□□□□□://□□□□□□.□□□/□□□□□□□□/□□□□□□□□

□□□□□://□□□□□□.□□□/□□□□□□801120/□□□□□□□□□□-□□-□□□□-□□□□-□□□□□□□□