

과제명 (Title) : 연구주제 및 방향 설정

일자	연구내용	지도 대학원생
10/16	<p><b>연구주제 및 방향 설정</b></p> <p><b>가진 능력</b></p> <ul style="list-style-type: none"> <li>- 파이썬(pandas, numpy, matplotlib, sklearn), R(기초, dplyr, ggplot...), 자바(기초)</li> <li>- Hadoop, Spark (기초), 쿼리문 기초</li> <li>- Spring Framework 통한 웹 구현(기초)</li> <li>- 학부수준의 통계지식(수리통계학, 회귀분석, 전산통계, 시계열.EDA)</li> <li>- kaggle, 공공데이터 포털 통한 데이터 확보 능력</li> </ul> <p><b>목표</b></p> <ul style="list-style-type: none"> <li>- 실제 데이터를 활용한 딥러닝 모델의 구현과 활용을 통한 딥러닝 방법의 이해와 실제적인 활용방안 탐구</li> </ul> <p><b>구체적인 활동</b></p> <p>1) 공모전 활동(하나에서 두개 선택해서 진행예정)</p> <p>1-1) MBN 빅데이터 아이디어</p> <ul style="list-style-type: none"> <li>- <a href="https://www.thinkcontest.com/Contest/ContestDetail.html?id=13084">https://www.thinkcontest.com/Contest/ContestDetail.html?id=13084</a></li> <li>- 샘플 데이터만 제공...</li> <li>- ~10/21까지(아이디어 계획서)</li> </ul> <p>1-2) 암(cancer) 빅데이터 활용 경진대회</p> <ul style="list-style-type: none"> <li>- <a href="https://www.thinkcontest.com/Contest/ContestDetail.html?id=13458">https://www.thinkcontest.com/Contest/ContestDetail.html?id=13458</a></li> <li>- 샘플 데이터만 제공</li> <li>- ~10/28까지(아이디어 계획서)</li> </ul> <p>1-3) L.point 빅데이터 컴퍼티션</p> <ul style="list-style-type: none"> <li>- <a href="https://competition.lpoint.com/front/Guideline.tran">https://competition.lpoint.com/front/Guideline.tran</a></li> <li>- 주제 : 디지털 행동기반 트렌드 예측</li> <li>- ~1/16까지(1차 제출)</li> </ul> <p>1-4) 대학생 인공지능 아이디어 경진대회</p> <p><a href="http://www.bitle.kr/board/view.bitle?boardId=BBS_00000006&amp;menuCd=DOM_000000104001000000&amp;orderBy=REGISTER_DATE%20DESC&amp;startPage=1&amp;dataSid=2076">http://www.bitle.kr/board/view.bitle?boardId=BBS_00000006&amp;menuCd=DOM_000000104001000000&amp;orderBy=REGISTER_DATE%20DESC&amp;startPage=1&amp;dataSid=2076</a></p> <ul style="list-style-type: none"> <li>- ~ 10/14일까지(아이디어 계획서)</li> </ul> <p>2) 자체적 활동과 결과물 산출</p> <p>2-1) 자체적으로 딥러닝 공부 후 결과 보고서 제작..</p>	
기록자:		(서명)

과제명 (Title) : 연구주제 구체화

일자

## 연구내용

지도  
대학원생

---

10/16

현재 지하철 광고패널은 **종일 같은 광고**를 띄우거나 **무작위적**으로 광고를 띄워 **효율이 낮은 방법**으로 광고를 하고 있다. 허나 디지털 패널광고의 경우, 시간대별로 간단히 **광고를 바꿔** 틀어줄 수 있으므로 분석결과를 활용하여 타겟팅 광고를 할 경우 **훨씬 효율적**이다.

서명



사진1. 지하철 디지털 광고 패널

## I. 유동인구 데이터의 활용

시간별 지역구별 유동인구 데이터를 통해, 원하는 지역(지하철역이 있는)의 시간대별 가장 많은 연령대를 파악하여, 해당 시간대의 광고 타겟층을 정함

TIME	CCORR1	CCORR2	CCORR3	CCORR4	CCORR5	CCORR6	CCORR7	CCORR8	CCORR9	CCORR10	CCORR11	CCORR12	CCORR13	CCORR14	CCORR15	CCORR16	CCORR17	CCORR18	CCORR19	CCORR20	CCORR21	CCORR22	CCORR23	CCORR24	CCORR25	CCORR26	CCORR27	CCORR28	CCORR29	CCORR30	CCORR31	CCORR32	CCORR33	CCORR34	CCORR35	CCORR36	CCORR37	CCORR38	CCORR39	CCORR40	CCORR41	CCORR42	CCORR43	CCORR44	CCORR45	CCORR46	CCORR47	CCORR48	CCORR49	CCORR50	CCORR51	CCORR52	CCORR53	CCORR54	CCORR55	CCORR56	CCORR57	CCORR58	CCORR59	CCORR60	CCORR61	CCORR62	CCORR63	CCORR64	CCORR65	CCORR66	CCORR67	CCORR68	CCORR69	CCORR70	CCORR71	CCORR72	CCORR73	CCORR74	CCORR75	CCORR76	CCORR77	CCORR78	CCORR79	CCORR80	CCORR81	CCORR82	CCORR83	CCORR84	CCORR85	CCORR86	CCORR87	CCORR88	CCORR89	CCORR90	CCORR91	CCORR92	CCORR93	CCORR94	CCORR95	CCORR96	CCORR97	CCORR98	CCORR99	CCORR100	CCORR101	CCORR102	CCORR103	CCORR104	CCORR105	CCORR106	CCORR107	CCORR108	CCORR109	CCORR110	CCORR111	CCORR112	CCORR113	CCORR114	CCORR115	CCORR116	CCORR117	CCORR118	CCORR119	CCORR120	CCORR121	CCORR122	CCORR123	CCORR124	CCORR125	CCORR126	CCORR127	CCORR128	CCORR129	CCORR130	CCORR131	CCORR132	CCORR133	CCORR134	CCORR135	CCORR136	CCORR137	CCORR138	CCORR139	CCORR140	CCORR141	CCORR142	CCORR143	CCORR144	CCORR145	CCORR146	CCORR147	CCORR148	CCORR149	CCORR150	CCORR151	CCORR152	CCORR153	CCORR154	CCORR155	CCORR156	CCORR157	CCORR158	CCORR159	CCORR160	CCORR161	CCORR162	CCORR163	CCORR164	CCORR165	CCORR166	CCORR167	CCORR168	CCORR169	CCORR170	CCORR171	CCORR172	CCORR173	CCORR174	CCORR175	CCORR176	CCORR177	CCORR178	CCORR179	CCORR180	CCORR181	CCORR182	CCORR183	CCORR184	CCORR185	CCORR186	CCORR187	CCORR188	CCORR189	CCORR190	CCORR191	CCORR192	CCORR193	CCORR194	CCORR195	CCORR196	CCORR197	CCORR198	CCORR199	CCORR200	CCORR201	CCORR202	CCORR203	CCORR204	CCORR205	CCORR206	CCORR207	CCORR208	CCORR209	CCORR210	CCORR211	CCORR212	CCORR213	CCORR214	CCORR215	CCORR216	CCORR217	CCORR218	CCORR219	CCORR220	CCORR221	CCORR222	CCORR223	CCORR224	CCORR225	CCORR226	CCORR227	CCORR228	CCORR229	CCORR230	CCORR231	CCORR232	CCORR233	CCORR234	CCORR235	CCORR236	CCORR237	CCORR238	CCORR239	CCORR240	CCORR241	CCORR242	CCORR243	CCORR244	CCORR245	CCORR246	CCORR247	CCORR248	CCORR249	CCORR250	CCORR251	CCORR252	CCORR253	CCORR254	CCORR255	CCORR256	CCORR257	CCORR258	CCORR259	CCORR260	CCORR261	CCORR262	CCORR263	CCORR264	CCORR265	CCORR266	CCORR267	CCORR268	CCORR269	CCORR270	CCORR271	CCORR272	CCORR273	CCORR274	CCORR275	CCORR276	CCORR277	CCORR278	CCORR279	CCORR280	CCORR281	CCORR282	CCORR283	CCORR284	CCORR285	CCORR286	CCORR287	CCORR288	CCORR289	CCORR290	CCORR291	CCORR292	CCORR293	CCORR294	CCORR295	CCORR296	CCORR297	CCORR298	CCORR299	CCORR300	CCORR301	CCORR302	CCORR303	CCORR304	CCORR305	CCORR306	CCORR307	CCORR308	CCORR309	CCORR310	CCORR311	CCORR312	CCORR313	CCORR314	CCORR315	CCORR316	CCORR317	CCORR318	CCORR319	CCORR320	CCORR321	CCORR322	CCORR323	CCORR324	CCORR325	CCORR326	CCORR327	CCORR328	CCORR329	CCORR330	CCORR331	CCORR332	CCORR333	CCORR334	CCORR335	CCORR336	CCORR337	CCORR338	CCORR339	CCORR340	CCORR341	CCORR342	CCORR343	CCORR344	CCORR345	CCORR346	CCORR347	CCORR348	CCORR349	CCORR350	CCORR351	CCORR352	CCORR353	CCORR354	CCORR355	CCORR356	CCORR357	CCORR358	CCORR359	CCORR360	CCORR361	CCORR362	CCORR363	CCORR364	CCORR365	CCORR366	CCORR367	CCORR368	CCORR369	CCORR370	CCORR371	CCORR372	CCORR373	CCORR374	CCORR375	CCORR376	CCORR377	CCORR378	CCORR379	CCORR380	CCORR381	CCORR382	CCORR383	CCORR384	CCORR385	CCORR386	CCORR387	CCORR388	CCORR389	CCORR390	CCORR391	CCORR392	CCORR393	CCORR394	CCORR395	CCORR396	CCORR397	CCORR398	CCORR399	CCORR400	CCORR401	CCORR402	CCORR403	CCORR404	CCORR405	CCORR406	CCORR407	CCORR408	CCORR409	CCORR410	CCORR411	CCORR412	CCORR413	CCORR414	CCORR415	CCORR416	CCORR417	CCORR418	CCORR419	CCORR420	CCORR421	CCORR422	CCORR423	CCORR424	CCORR425	CCORR426	CCORR427	CCORR428	CCORR429	CCORR430	CCORR431	CCORR432	CCORR433	CCORR434	CCORR435	CCORR436	CCORR437	CCORR438	CCORR439	CCORR440	CCORR441	CCORR442	CCORR443	CCORR444	CCORR445	CCORR446	CCORR447	CCORR448	CCORR449	CCORR450	CCORR451	CCORR452	CCORR453	CCORR454	CCORR455	CCORR456	CCORR457	CCORR458	CCORR459	CCORR460	CCORR461	CCORR462	CCORR463	CCORR464	CCORR465	CCORR466	CCORR467	CCORR468	CCORR469	CCORR470	CCORR471	CCORR472	CCORR473	CCORR474	CCORR475	CCORR476	CCORR477	CCORR478	CCORR479	CCORR480	CCORR481	CCORR482	CCORR483	CCORR484	CCORR485	CCORR486	CCORR487	CCORR488	CCORR489	CCORR490	CCORR491	CCORR492	CCORR493	CCORR494	CCORR495	CCORR496	CCORR497	CCORR498	CCORR499	CCORR500	CCORR501	CCORR502	CCORR503	CCORR504	CCORR505	CCORR506	CCORR507	CCORR508	CCORR509	CCORR510	CCORR511	CCORR512	CCORR513	CCORR514	CCORR515	CCORR516	CCORR517	CCORR518	CCORR519	CCORR520	CCORR521	CCORR522	CCORR523	CCORR524	CCORR525	CCORR526	CCORR527	CCORR528	CCORR529	CCORR530	CCORR531	CCORR532	CCORR533	CCORR534	CCORR535	CCORR536	CCORR537	CCORR538	CCORR539	CCORR540	CCORR541	CCORR542	CCORR543	CCORR544	CCORR545	CCORR546	CCORR547	CCORR548	CCORR549	CCORR550	CCORR551	CCORR552	CCORR553	CCORR554	CCORR555	CCORR556	CCORR557	CCORR558	CCORR559	CCORR560	CCORR561	CCORR562	CCORR563	CCORR564	CCORR565	CCORR566	CCORR567	CCORR568	CCORR569	CCORR570	CCORR571	CCORR572	CCORR573	CCORR574	CCORR575	CCORR576	CCORR577	CCORR578	CCORR579	CCORR580	CCORR581	CCORR582	CCORR583	CCORR584	CCORR585	CCORR586	CCORR587	CCORR588	CCORR589	CCORR590	CCORR591	CCORR592	CCORR593	CCORR594	CCORR595	CCORR596	CCORR597	CCORR598	CCORR599	CCORR600	CCORR601	CCORR602	CCORR603	CCORR604	CCORR605	CCORR606	CCORR607	CCORR608	CCORR609	CCORR610	CCORR611	CCORR612	CCORR613	CCORR614	CCORR615	CCORR616	CCORR617	CCORR618	CCORR619	CCORR620	CCORR621	CCORR622	CCORR623	CCORR624	CCORR625	CCORR626	CCORR627	CCORR628	CCORR629	CCORR630	CCORR631	CCORR632	CCORR633	CCORR634	CCORR635	CCORR636	CCORR637	CCORR638	CCORR639	CCORR640	CCORR641	CCORR642	CCORR643	CCORR644	CCORR645	CCORR646	CCORR647	CCORR648	CCORR649	CCORR650	CCORR651	CCORR652	CCORR653	CCORR654	CCORR655	CCORR656	CCORR657	CCORR658	CCORR659	CCORR660	CCORR661	CCORR662	CCORR663	CCORR664	CCORR665	CCORR666	CCORR667	CCORR668	CCORR669	CCORR670	CCORR671	CCORR672	CCORR673	CCORR674	CCORR675	CCORR676	CCORR677	CCORR678	CCORR679	CCORR680	CCORR681	CCORR682	CCORR683	CCORR684	CCORR685	CCORR686	CCORR687	CCORR688	CCORR689	CCORR690	CCORR691	CCORR692	CCORR693	CCORR694	CCORR695	CCORR696	CCORR697	CCORR698	CCORR699	CCORR700	CCORR701	CCORR702	CCORR703	CCORR704	CCORR705	CCORR706	CCORR707	CCORR708	CCORR709	CCORR710	CCORR711	CCORR712	CCORR713	CCORR714	CCORR715	CCORR716	CCORR717	CCORR718	CCORR719	CCORR720	CCORR721	CCORR722	CCORR723	CCORR724	CCORR725	CCORR726	CCORR727	CCORR728	CCORR729	CCORR730	CCORR731	CCORR732	CCORR733	CCORR734	CCORR735	CCORR736	CCORR737	CCORR738	CCORR739	CCORR740	CCORR741	CCORR742	CCORR743	CCORR744	CCORR745	CCORR746	CCORR747	CCORR748	CCORR749	CCORR750	CCORR751	CCORR752	CCORR753	CCORR754	CCORR755	CCORR756	CCORR757	CCORR758	CCORR759	CCORR760	CCORR761	CCORR762	CCORR763	CCORR764	CCORR765	CCORR766	CCORR767	CCORR768	CCORR769	CCORR770	CCORR771	CCORR772	CCORR773	CCORR774	CCORR775	CCORR776	CCORR777	CCORR778	CCORR779	CCORR780	CCORR781	CCORR782	CCORR783	CCORR784	CCORR785	CCORR786	CCORR787	CCORR788	CCORR789	CCORR790	CCORR791	CCORR792	CCORR793	CCORR794	CCORR795	CCORR796	CCORR797	CCORR798	CCORR799	CCORR800	CCORR801	CCORR802	CCORR803	CCORR804	CCORR805	CCORR806	CCORR807	CCORR808	CCORR809	CCORR810	CCORR811	CCORR812	CCORR813	CCORR814	CCORR815	CCORR816	CCORR817	CCORR818	CCORR819	CCORR820	CCORR821	CCORR822	CCORR823	CCORR824	CCORR825	CCORR826	CCORR827	CCORR828	CCORR829	CCORR830	CCORR831	CCORR832	CCORR833	CCORR834	CCORR835	CCORR836	CCORR837	CCORR838	CCORR839	CCORR840	CCORR841	CCORR842	CCORR843	CCORR844	CCORR845	CCORR846	CCORR847	CCORR848	CCORR849	CCORR850	CCORR851	CCORR852	CCORR853	CCORR854	CCORR855	CCORR856	CCORR857	CCORR858	CCORR859	CCORR860	CCORR861	CCORR862	CCORR863	CCORR864	CCORR865	CCORR866	CCORR867	CCORR868	CCORR869	CCORR870	CCORR871	CCORR872	CCORR873	CCORR874	CCORR875	CCORR876	CCORR877	CCORR878	CCORR879	CCORR880	CCORR881	CCORR882	CCORR883	CCORR884	CCORR885	CCORR886	CCORR887	CCORR888	CCORR889	CCORR890	CCORR891	CCORR892	CCORR893	CCORR894	CCORR895	CCORR896	CCORR897	CCORR898	CCORR899	CCORR900	CCORR901	CCORR902	CCORR903	CCORR904	CCORR905	CCORR906	CCORR907	CCORR908	CCORR909	CCORR910	CCORR911	CCORR912	CCORR913	CCORR914	CCORR915	CCORR916	CCORR917	CCORR918	CCORR919	CCORR920	CCORR921	CCORR922	CCORR923	CCORR924	CCORR925	CCORR926	CCORR927	CCORR928	CCORR929	CCORR930	CCORR931	CCORR932	CCORR933	CCORR934	CCORR935	CCORR936	CCORR937	CCORR938	CCORR939	CCORR940	CCORR941	CCORR942	CCORR943	CCORR944	CCORR945	CCORR946	CCORR947	CCORR948	CCORR949	CCORR950	CCORR951	CCORR952	CCORR953	CCORR954	CCORR955	CCORR956	CCORR957	CCORR958	CCORR959	CCORR960	CCORR961	CCORR962	CCORR963	CCORR964	CCORR965	CCORR966	CCORR967	CCORR968	CCORR969	CCORR970	CCORR971	CCORR972	CCORR973	CCORR974	CCORR975	CCORR976	CCORR977	CCORR978	CCORR979	CCORR980	CCORR981	CCORR982	CCORR983	CCORR984	CCORR985	CCORR986	CCORR987	CCORR988	CCORR989	CCORR990	CCORR991	CCORR992	CCORR993	CCORR994	CCORR995	CCORR996	CCORR997	CCORR998	CCORR999	CCORR1000
------	--------	--------	--------	--------	--------	--------	--------	--------	--------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	-----------

사진 2. 시간대별 유동인구 데이터

## II. 시계열 데이터의 분석

01~24 시까지 시간별로 가장 많은 연령층이 무엇인지 파악하기 위해 시간대별로 변화패턴을 보이는 시계열자료를 예측하기 위한 모델이 필요하다.

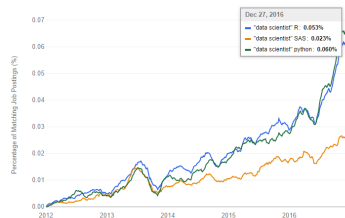


사진 3. 시계열 자료

### III. 시계열 예측을 위한 LSTM 모델

RNN 모델(순환신경망 모델)의 발전된 형태인 LSTM(Long Short Term Memory networks) 모델은 이러한 시계열 자료의 예측에 뛰어난 성능을 보인다. 따라서 LSTM을 이용하여 시계열자료인 유동인구데이터를 분석, 예측하여 이를 기반으로 해당시간의 광고 타겟층을 파악한다.

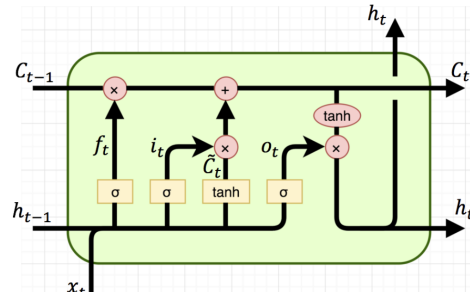


사진 4. LSTM 모델의 구조

### IV. 모델의 작동 과정

#### 1) 데이터 수집

- API를 통한 데이터수집과 Feature engineering 과정을 파이썬 모듈화 하여 주기적으로 크롤링한다.

#### 2) LSTM 모델의 학습

- api를 통한 데이터수집과 Feature engineering 과정을 파이썬 모듈화 하여 주기적으로 크롤링한다.
- 데이터를 이용하여 모델을 학습시킨다.

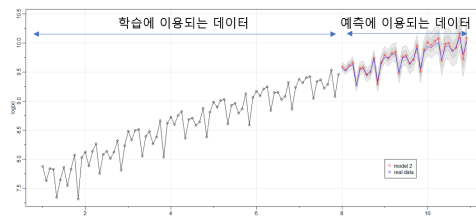


사진 5. 학습과정

### V. 학습된 모델 통한 예측 및 광고 추천

학습된 모델을 통해 예측된 유동인구를 바탕으로 해당 시간에 가장 많은 연령층을 타겟으로 맞춤 광고를 추천

08시	09시	10시	11시
10대 남 우세 게임, 공부	20대 여 우세 화장품	40대 남 우세 은행 상품	60대 남녀 우세 여행 패키지

표 1. 결과의 예시

기록자:

(서명)

과제명 (Title) : LSTM paper review

일자

연구내용

지도  
대학원생

10/23

## A Critical Review of Recurrent Neural Networks for Sequence Learning

Zachary C. Lipton  
zlipton@cs.ucsd.edu

John Berkowitz  
jaberkow@physics.ucsd.edu

Charles Elkan  
elkan@cs.ucsd.edu

June 5th, 2015

## Sequence 형태의 데이터를 위한 RNN model

### Introduction

#### Why not use markov models

연산량이 너무 많다.

각 간격마다, 'state space'(상태공간)가 필요하다.  
Seq가 길어질수록 알고리즘의 시간복잡도가  
크게 증가  $\Rightarrow O(N^2 T)$

$$P = \begin{pmatrix} 0 & \frac{4}{5} & 0 & \frac{1}{5} & 0 \\ \frac{1}{4} & 0 & \frac{1}{2} & \frac{1}{4} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{10} & \frac{2}{5} \\ 0 & 0 & 0 & 1 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 \end{pmatrix}$$

$$x_t \xrightarrow{p(\text{전이행렬})} x_{t+1}$$

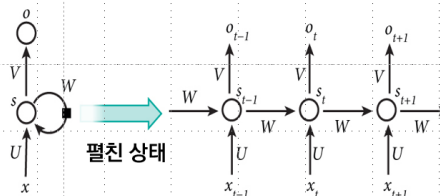
### Recurrent neural network

#### RNN의 학습

#### BPTT

Backpropagation through time

기존의 feedforward 방식의 신경망에  
적용되는 'Backpropagation' 알고리즘을  
펼친 상태의 RNN신경망에 동일하게 적용



기록자:

(서명)

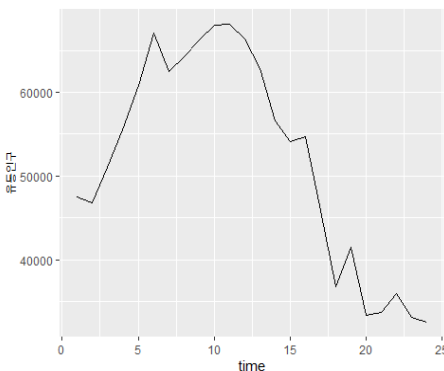
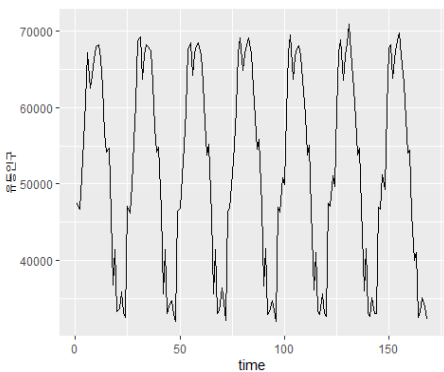
과제명 (Title) : LSTM paper review-2

일자	연구내용	지도 대학원생									
10/30	<p>Recurrent neural network</p> <p>문제점</p> <p>Issue of local optima</p> <div data-bbox="469 636 644 792"> </div> <p>n이 커짐에 따라 error의 Saddle point의 수가 기하급수적으로 증가</p> <div data-bbox="724 703 836 781"> </div> <div data-bbox="868 613 1155 781"> </div> <p>따라서 지역해에 수렴하기 쉬워져 학습속도가 매우 느려진다.</p> <p>Modern RNN architectures</p> <p>LSTM(Long short-term memory)</p> <p>★ 'memory cell'이라는 개념을 도입하여 'vanishing gradient' 문제를 해결하여 긴term의 연관성을 가진 data를 학습시키기에 유리하다</p> <div data-bbox="564 1151 1155 1386"> </div> <p>Modern RNN architectures</p> <p>Memory cell의 node와 gate</p> <table border="1"> <thead> <tr> <th>Input node <math>g_c</math></th><th>Internal state <math>s_c</math></th><th>Forget gate <math>f_c</math></th></tr> </thead> <tbody> <tr> <td><math>x_t</math>와 <math>h_{t-1}</math>을 받은 뒤, tanh 활성화함수를 거치는 구간</td><td><math>x_t</math>와 <math>i_t</math>를 원소끼리 곱해준 뒤, 직전값 (<math>s_{t-1}</math>, 이를 recurrent edge를 가졌다고 표현)을 더해준다.</td><td>Internal state를 계산할 때, <math>s_{t-1}</math>에 곱해져 이전의 기억들을 잊게 해주는 역할을 한다. 계속해서 새로운 데이터를 학습해야 하는 경우 유용하다</td></tr> <tr> <td>Input gate <math>i_c</math> <math>x_t</math>와 <math>h_{t-1}</math>을 받은 뒤, sigmoid 활성화함수를 거치기 때문에 0~1사이의 값을 가진다. Input node 값에 곱해져 신호를 자르거나 보내주는 gate의 역할을 한다</td><td>이때 <math>W=1</math>을 고정적으로 가지기 때문에 'gradient vanishing' 문제를 보이지 않는다.</td><td>Output gate <math>o_c</math> Memory cell의 최종적인 출력값을 내보낼 때, <math>s_t</math>에 곱해진다. 이때 별도의 활성화함수를 거치지 않는다.</td></tr> </tbody> </table>	Input node $g_c$	Internal state $s_c$	Forget gate $f_c$	$x_t$ 와 $h_{t-1}$ 을 받은 뒤, tanh 활성화함수를 거치는 구간	$x_t$ 와 $i_t$ 를 원소끼리 곱해준 뒤, 직전값 ( $s_{t-1}$ , 이를 recurrent edge를 가졌다고 표현)을 더해준다.	Internal state를 계산할 때, $s_{t-1}$ 에 곱해져 이전의 기억들을 잊게 해주는 역할을 한다. 계속해서 새로운 데이터를 학습해야 하는 경우 유용하다	Input gate $i_c$ $x_t$ 와 $h_{t-1}$ 을 받은 뒤, sigmoid 활성화함수를 거치기 때문에 0~1사이의 값을 가진다. Input node 값에 곱해져 신호를 자르거나 보내주는 gate의 역할을 한다	이때 $W=1$ 을 고정적으로 가지기 때문에 'gradient vanishing' 문제를 보이지 않는다.	Output gate $o_c$ Memory cell의 최종적인 출력값을 내보낼 때, $s_t$ 에 곱해진다. 이때 별도의 활성화함수를 거치지 않는다.	
Input node $g_c$	Internal state $s_c$	Forget gate $f_c$									
$x_t$ 와 $h_{t-1}$ 을 받은 뒤, tanh 활성화함수를 거치는 구간	$x_t$ 와 $i_t$ 를 원소끼리 곱해준 뒤, 직전값 ( $s_{t-1}$ , 이를 recurrent edge를 가졌다고 표현)을 더해준다.	Internal state를 계산할 때, $s_{t-1}$ 에 곱해져 이전의 기억들을 잊게 해주는 역할을 한다. 계속해서 새로운 데이터를 학습해야 하는 경우 유용하다									
Input gate $i_c$ $x_t$ 와 $h_{t-1}$ 을 받은 뒤, sigmoid 활성화함수를 거치기 때문에 0~1사이의 값을 가진다. Input node 값에 곱해져 신호를 자르거나 보내주는 gate의 역할을 한다	이때 $W=1$ 을 고정적으로 가지기 때문에 'gradient vanishing' 문제를 보이지 않는다.	Output gate $o_c$ Memory cell의 최종적인 출력값을 내보낼 때, $s_t$ 에 곱해진다. 이때 별도의 활성화함수를 거치지 않는다.									
기록자:		(서명)									

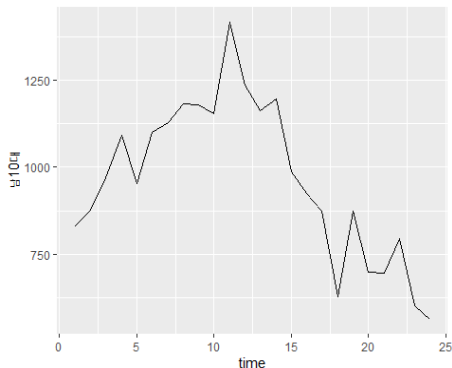
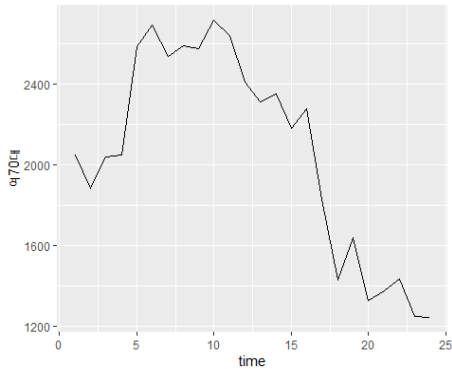
과제명 (Title) : 데이터 전처리-1

일자	연구내용	지도 대학원생
11/2	<p>simulation 100jy 2019 11 2 전체소스코드 : <a href="https://100jy.github.io/1stm_com/%EB%A7%88%ED%81%AC%EB%8B%A4%EC%9A%B4">https://100jy.github.io/1stm_com/%EB%A7%88%ED%81%AC%EB%8B%A4%EC%9A%B4</a></p> <p><b>가. data simulation(유동인구)</b> data simulation... setwd("C:/Users/wnduq/Desktop") data1&lt;-read.csv("월평균 시간대별 유동인구.csv") data2&lt;-read.csv("월평균 연령별 유동인구.csv") data3&lt;-read.csv("월평균 요일별 유동인구.csv") data4&lt;-read.csv("월평균 성별 유동인구.csv")</p> <p>##부전1동 data_time&lt;-data1[which(data1\$분석영역=="부전1동"),] data_old&lt;-data2[which(data2\$분석영역=="부전1동"),] data_days&lt;-data3[which(data3\$읍면동=="부전제1동"),] data_sex&lt;-data4[which(data4\$읍면동=="부전제1동"),] #####</p> <p>###16년 화요일 missing(보간법) weds&lt;-round((data_days[which(data_days\$년==2016&amp;data_days\$요일=="월"),]\$평균.유동인구 + data_days[which(data_days\$년==2016&amp;data_days\$요일=="수"),]\$평균.유동인구)/2)</p> <p>년=rep(2016,12) 구군=rep("부산진구",12) 읍명동=rep("부전제1동",12) 요일="화" 월=1.12 주중.주말=rep("주중",12) 평균.유동인구=weds add=cbind(년,구군,읍명동,요일,월,주중.주말,평균.유동인구) colnames(add)&lt;-colnames(data_days)</p> <p>data_days=rbind(data_days,add)</p> <p>##### (중략) #####</p> <p>factor=factor(rep(월,4),levels=name) ####날짜 달아주기 ##여기서부터 문제있음... n=25536 시간=rep(rep(0.23,rep(7,24)),n/(7*24)) 주차=rep(1.4,c(n/4,n/4,n/4,n/4)) 요일=rep(1.7,n/7) 부전1동_유동_plat=cbind(c(tmp[,1],tmp[,2],tmp[,3],tmp[,4]),rep(월,4),주차,시간,요일) dim(부전1동_유동_plat) ## [1] 25536 5 #24*28*(12*3+2) uo=부전1동_유동_plat uo=as.data.frame(uo) str(uo) ## 'data.frame': 25536 obs. of 5 variables: ## \$ V1 : Factor w/ 25536 levels "14140.6349032871",...: 9755 9614 9370 9378 9557 9717 9559 9486 9282 9507 ... ## ..- attr(*, "names")= chr "" "" "" "" "" ... ## \$ V2 : Factor w/ 38 levels "201601","201602",...: 1 1 1 1 1 1 1 1 1 1 1 ... ## ..- attr(*, "names")= chr "" "" "" "" "" ... ## \$ 주차: Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 1 ... ## ..- attr(*, "names")= chr "" "" "" "" "" ... ## \$ 시간: Factor w/ 24 levels "0","1","10","11",...: 1 1 1 1 1 1 1 1 2 2 2 ... ## ..- attr(*, "names")= chr "" "" "" "" "" ... ## \$ 요일: Factor w/ 7 levels "1","2","3","4",...: 1 2 3 4 5 6 7 1 2 3 ... ## ..- attr(*, "names")= chr "" "" "" "" "" ... uo\$V2=factor uo\$시간=as.integer(as.character(uo\$시간)) 부전1동_유동_plat=uo%&gt;% arrange(V2,주차,요일,시간) colnames(부전1동_유동_plat)[1]="유동인구" 부전1동_유동_plat\$유동인구=as.integer(as.character(부전1동_유동_plat\$유동인구)) time=1.25536 부전1동_유동_plat\$time=time 24*12*28 ## [1] 8064</p>	
기록자:		(서명)

과제명 (Title) : 데이터 전처리-2

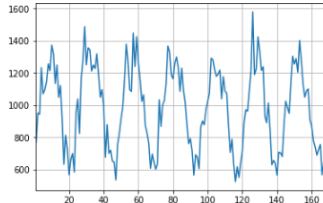
일자	연구내용	지도 대학원생
11/6	<p>simulation 100jy 2019 11 6 전체소스코드 : <a href="https://100jy.github.io/lstm_com/%EB%A7%88%ED%81%AC%EB%88%A4%EC%9A%B4">https://100jy.github.io/lstm_com/%EB%A7%88%ED%81%AC%EB%88%A4%EC%9A%B4</a></p> <pre> x11() library(ggplot2) ### 하루치 x11() ggplot(부전1동_유동_plat[1:24,], aes(x=time, y=유동인구)) +   geom_line() </pre>  <pre> ## 일주일치 x11() ggplot(부전1동_유동_plat[1:(24*7),], aes(x=time, y=유동인구)) +   geom_line() </pre> 	
기록자:		(서명)

과제명 (Title) : 데이터 전처리-3

일자	연구내용	지도 대학원생
11/12	<p>simulation 100jy</p> <p>전체소스코드 : <a href="https://100jy.github.io/lstm_com/%EB%A7%88%ED%81%AC%EB%8B%A4%EC%9A%B4">https://100jy.github.io/lstm_com/%EB%A7%88%ED%81%AC%EB%8B%A4%EC%9A%B4</a></p> <pre>##### ###도그바와 해주며 줄을 두 ###새벽 4 5시에 사람이 잠은???? #####성별여령별 걸려 나누기 ###일단위로 숫자붙은뒤 비율로 변환... ###일단위로 랜덤하게 뽑기 ###노이즈 섞어서 늘리고 이후에 비율화. data sex&lt;-data4[which(data4\$음면동=="부전제1동"),] data old&lt;-data2[which(data2\$부전지역=="부전1동"),] data old=data old[-c(1:27),] ##### data sex\$남자.전체.유동인구=as.integer(as.character(data sex\$남자.전체.유동인구)) data sex\$여자.전체.유동인구=as.integer(as.character(data sex\$여자.전체.유동인구)) data sex\$남성비율=(data sex\$남자.전체.유동인구)/((data sex\$남자.전체.유동인구)+(data sex\$여자.전체.유동인구)) data sex\$여성비율=rep(1,38)-(data sex\$남성비율) p1=cbind(data sex\$남성비율,data sex\$여성비율) p1=cbind(p1,p1,p1,p1,p1,p1) tmp=cbind(data old[,3],data old[,4],data old[,5],data old[,6],data old[,7],data old[,8],data old[,9],data old[,10]) data old=cbind(data old[,c(1:2)],tmp) colnames(data old)[c(3:16)]=c("10대남","10대여","20대남","20대여","30대남","30대여", "40대남","40대여","50대남","50대여","60대남","60대여", "70대남","70대여") ##### gnor=function(x){   rnorm(14*24*28,rep(as.integer(x[-c(1:2)]),rep(24*28,14)),50) } tmp=apply(data old,MARGIN = 1,FUN = gnor) dim(tmp)  ## [1] 9408 38  ####(중략)  colnames(부전1동 유동 last)=c("남10대","여10대","이십대남","이십대여","30대남","30대여", "40대남","40대여","50대남","50대여","60대남","60대여", "70대남","70대여","time") 부전1동 유동 last=as.data.frame(부전1동 유동 last) ###하루치(10대남) x11() ggplot(부전1동 유동 last[1:24,], aes(x=time, y= 남10대)) +   geom_line()</pre>  <pre>x11() ggplot(부전1동 유동 last[1:24,], aes(x=time, y= 여70대)) +   geom_line()</pre> 	
기록자:		(서명)



## 과제명 (Title) : 모델링

일자	연구내용	지도 대학원생
11/17	<p>전체 소스코드 : <a href="https://github.com/100jy/lstm_com/blob/master/pred_people_simple_ver.ipynb">https://github.com/100jy/lstm_com/blob/master/pred_people_simple_ver.ipynb</a></p> <p><b>LSTM모델을 통한 유동인구 예측(simple_ver.)</b></p> <p>간단한 구조의 LSTM model을 활용하여 유동인구 데이터를 예측</p> <pre> In [ ]: import matplotlib.pyplot as plt import tensorflow as tf import numpy as np import pandas as pd import os from sklearn.preprocessing import MinMaxScaler  In [31]: from tensorflow.python.keras.layers import LSTM from tensorflow.python.keras.models import Sequential from tensorflow.python.keras.layers import Dense from tensorflow.python.keras import backend as K from tensorflow.python.keras import optimizers from tensorflow.python.keras.callbacks import EarlyStopping, ModelCheckpoint, TensorBoard, ReduceLROnPlateau  In [50]: df=pd.read_csv("C:/Users/wndu/Desktop/유동인구_부천1동.csv",encoding="euc-kr") df.shape  Out [50]: (25536, 17)  In [ ]: df["남10대"].plot() df["남50대"].plot() df["여20대"].plot()  In [51]: #일주일치 df["남10대"][1:24*7].plot(grid=True)  Out [51]: &lt;matplotlib.axes._subplots.AxesSubplot at 0x218c69e4f60&gt; </pre>  <p>실적 과적합되는 느낌... (1. 배치사이즈 조절 2. ...)</p> <pre> In [63]: ##데이터 #scaling scaler = MinMaxScaler(feature_range=(0, 1)) tmp_data = scaler.fit_transform(df["남10대"][1:None])  #일주일치 1시간 예측(슬라이드 윈도우) look_back = 24*2  def create_dataset(dataset, look_back):     dataX, dataY = [], []     for i in range(len(dataset)-look_back-1):         a = dataset[i:(i + look_back)]         dataX.append(a)         dataY.append(dataset[i + look_back])     return np.array(dataX), np.array(dataY)  data={} data=create_dataset(tmp_data, look_back)  x_data=data[0] y_data=data[1] num_data = len(x_data)  ##test,train 분할(1번치만 학습)  num_train = 24*28*24 num_test = 24*28*1  x_train = x_data[0:num_train] x_test = x_data[num_train:num_train+num_test-1]  y_train = y_data[0:num_train] y_test = y_data[num_train:num_train+num_test-1]  num_x_signals = x_data.shape[1] num_x_signals  num_y_signals = 1 num_y_signals </pre>	
기록자:		(서명)

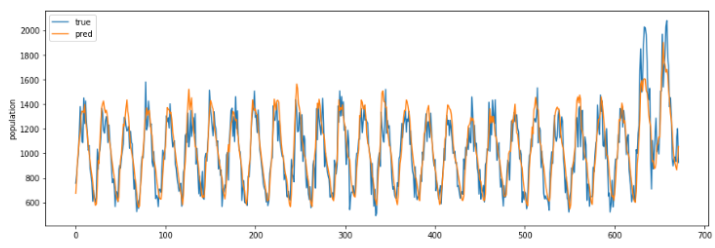
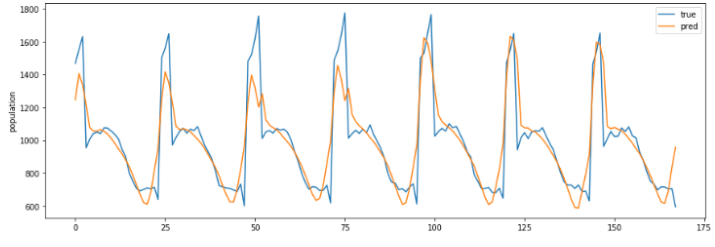
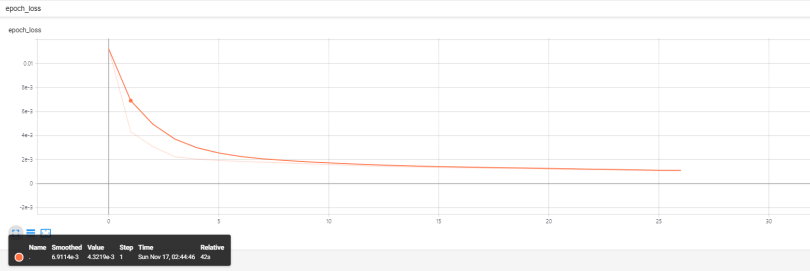
과제명 (Title) : 모델링-2

일자	연구내용	지도 대학원생
11/20	<div>전체 소스코드 : <a href="https://github.com/100yy/lstm_com/blob/master/pred_people_simple_ver.ipynb">https://github.com/100yy/lstm_com/blob/master/pred_people_simple_ver.ipynb</a></div> <div><pre>##모델링(simple_ver.)  K.clear_session()  model = Sequential() # Sequential Model model.add(LSTM(20, input_shape=(24+2, 1))) # (timestamp, feature) model.add(Dense(1)) # output = 1 model.compile(loss='mean_squared_error', optimizer='adam') model.summary()  x_train = x_train.reshape(x_train.shape[0],24+2,1)  ##callback early_stop = EarlyStopping(monitor='loss', patience=1, verbose=1) callback_tensorboard = TensorBoard(log_dir='./last_M/', histogram_freq=0, write_graph=True)  model.fit(x_train, y_train, epochs=100,         batch_size=50, verbose=1, callbacks=[early_stop, callback_tensorboard])  Model: "sequential" ----- Layer (type)                 Output Shape              Param # ----- lstm (LSTM)                  (None, 20)                1760 ----- dense (Dense)                (None, 1)                 21 ----- Total params: 1,781 Trainable params: 1,781 Non-trainable params: 0 ----- Epoch 1/100 16128/16128 [=====] - 42s 3ms/sample - loss: 0.0112 Epoch 2/100 16128/16128 [=====] - 43s 3ms/sample - loss: 0.0043 Epoch 3/100 16128/16128 [=====] - 45s 3ms/sample - loss: 0.0031 Epoch 4/100 16128/16128 [=====] - 42s 3ms/sample - loss: 0.0022 Epoch 5/100 16128/16128 [=====] - 42s 3ms/sample - loss: 0.0020 Epoch 6/100  학습이후 예측값과 실제값을 비교하여 본다.  그림을 통한 비교  In [70]: ##예측  def plot_comparison(start_idx, length, train):      if train:         x = x_train         y_true = y_train     else:         x = x_test.reshape(x_test.shape[0],24+2,1)         y_true = y_test     end_idx = start_idx + length     x = x[start_idx:end_idx]     y_true = y_true[start_idx:end_idx]     y_pred = model.predict(x)      signal_pred = scaler.inverse_transform(y_pred)     signal_true = scaler.inverse_transform(y_true)     plt.figure(figsize=(15,5))     plt.plot(signal_true, label='true')     plt.plot(signal_pred, label='pred')      plt.ylabel("population")     plt.legend()     plt.show()  ##training set에 대해 plot_comparison(0, length=24+28, train=True)</pre></div>	

기록자:

(서명)

과제명 (Title) : 모델링-3

일자	연구내용	지도 대학원생
11/24	<p>전체 소스코드 : <a href="https://github.com/100yy/lstm_com/blob/master/pred_people_simple_ver.ipynb">https://github.com/100yy/lstm_com/blob/master/pred_people_simple_ver.ipynb</a></p>  <p>training set에 대한 적합을 본다.</p> <pre>In [69]: ###predict plot_comparison(0, length=24*7, train=False)</pre>  <p>test set에 대한 predict값과 실제값을 비교하여 본다.</p> 	

기록자:

(서명)

## 과제명 (Title) : 모델링-4

일자

연구내용

지도  
대학원생

11/27

전체 소스코드 : [https://github.com/100y/lstm\\_com/blob/master/pred\\_people\\_simple\\_ver.ipynb](https://github.com/100y/lstm_com/blob/master/pred_people_simple_ver.ipynb)

### import modules

```
In [1]: import matplotlib.pyplot as plt
import tensorflow as tf
import numpy as np
import pandas as pd
import os
from sklearn.preprocessing import MinMaxScaler

In [2]: from tensorflow.python.keras.layers import LSTM
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dense
from tensorflow.python.keras import backend as K
from tensorflow.python.keras import optimizers
from tensorflow.python.keras.callbacks import EarlyStopping, ModelCheckpoint, TensorBoard, ReduceLROnPlateau
```

### 유동인구 data 불러오기

```
In [16]: df=pd.read_csv('C:/Users/wndu/Desktop/유동인구_최종/유동인구_부정1출.csv',encoding='euc-kr')
df

Out[16]:
```

	Unnamed: 0	남10대	여10대	남20대	여20대	남30대	여30대	남40대
0	1	905.809787	887.335419	4223.613759	4277.240753	4180.953422	4115.471999	4088.888921
1	2	915.703151	826.247217	4280.190893	4325.748017	4017.522110	4218.581833	4181.859972
2	3	931.880078	1219.284692	4484.528978	4539.528097	4431.449251	4490.087514	4187.246254
3	4	1082.323717	1134.101183	4988.850871	4997.978477	4934.084081	4870.484613	4540.073216
4	5	1300.297881	975.982514	5307.215472	5343.285054	5183.591120	5131.022234	5188.881198
...	...	...	...	...	...	...	...	...
25531	25532	942.032794	982.913584	3993.288317	4052.804238	3998.428185	3781.843405	3458.584887
25532	25533	918.533195	940.288290	3857.985039	3947.819787	3801.931740	3684.245235	3384.934030
25533	25534	755.580838	773.282828	3174.366754	3246.801482	2982.212012	3028.732201	2769.874894
25534	25535	848.474448	883.432829	2722.174284	2785.433673	2540.485927	2598.178705	2373.822485
25535	25536	581.792929	574.873690	2383.388697	2415.889788	2204.084138	2254.884874	2080.085887

25536 rows x 17 columns

### 입력데이터,출력데이터 생성

48시간 간격의 sliding window 생성, minmax scaling으로 학습속도 높인다.

```
In [17]: #feature생성..

scaler = MinMaxScaler(feature_range=(0, 1))
df = pd.DataFrame(scaler.fit_transform(np.array(df.iloc[:,1:16])))

def create_dataset(dataset, look_back):
    total=[]
    x_data=pd.DataFrame(np.zeros((25487,48)))
    y_data=pd.DataFrame(np.zeros((25487,1)))

    for j in range(14) :
        tmp_data = np.array(dataset.iloc[:,j:j+1])
        dataX, dataY = [], []
        for i in range(len(tmp_data)-look_back-1):
            a = tmp_data[i:(i + look_back)]
            dataX.append(a)
            dataY.append(tmp_data[i + look_back])

        x=pd.DataFrame((np.array(dataX).reshape(25487,48)))
        y=pd.DataFrame((np.array(dataY).reshape(25487,1)))
        x_data=pd.concat([x_data,x],axis=1)
        y_data=pd.concat([y_data,y],axis=1)

    return x_data,y_data

#2일치로 1시간 예측(슬라이드 윈도우)
look_back = 24*2
temp=create_dataset(df,look_back)
x_data=temp[0]
y_data=temp[1]

x_data.columns=range(720)
y_data.columns=range(15)

x_data=np.array(x_data.drop(x_data.columns[0:48], axis='columns'))
y_data=np.array(y_data.drop(y_data.columns[0], axis='columns'))
x_data.shape,y_data.shape

Out[17]: ((25487, 672), (25487, 14))
```

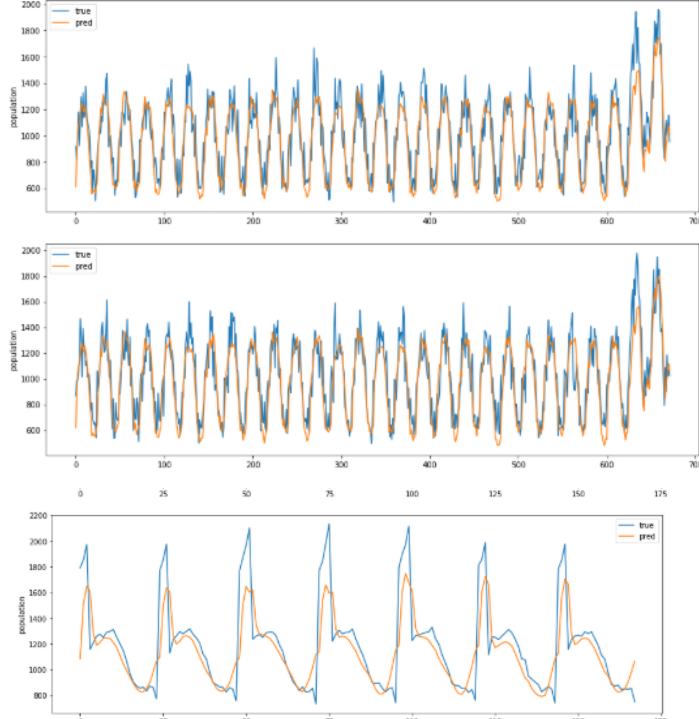
기록자:

(서명)

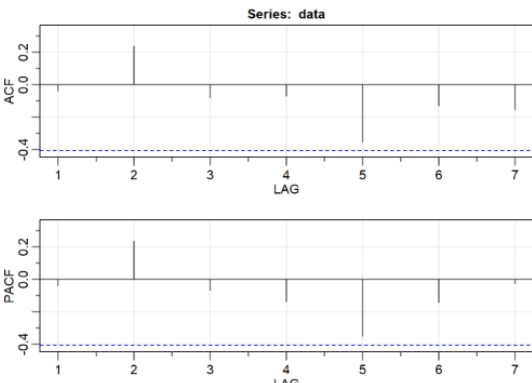
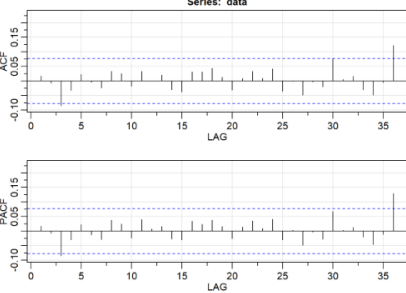
## 과제명 (Title) : 모델링-5

일자	연구내용	지도 대학원생
11/30	<p>전체 소스코드 : <a href="https://github.com/100y/lstm_com/blob/master/pred_people_simple_ver.ipynb">https://github.com/100y/lstm_com/blob/master/pred_people_simple_ver.ipynb</a></p> <p><b>모델생성 및 학습</b></p> <p>trainset(2년),testset(2개월)으로 LSTM모델학습. 배치당 20분소요...</p> <pre> In [19]: ##test,train 분할(2년치)만 학습)  num_train = 24*28*24 num_test = 24*28*2  x_train = x_data[0:num_train] x_test = x_data[num_train:num_train+num_test-1]  y_train = y_data[0:num_train] y_test = y_data[num_train:num_train+num_test-1]  num_x_signals = x_data.shape[1] num_x_signals  num_y_signals = 1 num_y_signals  ##모델링(simple_ver.)  K.clear_session()  model = Sequential() # Sequential Model model.add(LSTM(512, input_shape=(24*2*14,1))) # {timestep+feature,1} model.add(Dense(14)) # output = 14 model.compile(loss='mean_squared_error', optimizer='adam') model.summary()  x_train = x_train.reshape(x_train.shape[0],24*2*14,1)  ##callback early_stop = EarlyStopping(monitor='loss', patience=1, verbose=1) callback_tensorboard = TensorBoard(log_dir='./last_tf', histogram_freq=0, write_graph=True)  model.fit(x_train, y_train, epochs=100,         batch_size=50, verbose=1, callbacks=[early_stop,callback_tensorboard])#gpu메모리할부족 batch_size=5 0타게...  .. . . .  Model: "sequential" Layer (type) Output Shape Param # ===== lstm (LSTM) (None, 512) 1052872 ----- dense (Dense) (None, 14) 7182 ----- Total params: 1,059,854 Trainable params: 1,059,854 Non-trainable params: 0 ----- Epoch 1/100 18128/18128 [=====] - 1812s 81ms/sample - loss: 0.0189 Epoch 2/100 18128/18128 [=====] - 1297s 80ms/sample - loss: 0.0041 Epoch 3/100 18128/18128 [=====] - 1297s 80ms/sample - loss: 0.0028 Epoch 4/100 18128/18128 [=====] - 1298s 80ms/sample - loss: 0.0024 Epoch 5/100 18128/18128 [=====] - 1297s 80ms/sample - loss: 0.0028 Epoch 6/100 18128/18128 [=====] - 1299s 81ms/sample - loss: 0.0021 Epoch 7/100 18128/18128 [=====] - 1800s 81ms/sample - loss: 0.0019 Epoch 8/100 18128/18128 [=====] - 1297s 80ms/sample - loss: 0.0016 Epoch 9/100 18128/18128 [=====] - 1297s 80ms/sample - loss: 0.0016 Epoch 10/100 18128/18128 [=====] - 1298s 80ms/sample - loss: 0.0015 Epoch 11/100 18128/18128 [=====] - 1297s 80ms/sample - loss: 0.0018 Epoch 12/100 18128/18128 [=====] - 1295s 80ms/sample - loss: 0.0014 Epoch 00012: early stopping 19]: &lt;tensorflow.python.keras.callbacks.History at 0x25a28c283c8&gt; </pre>	
기록자:		(서명)

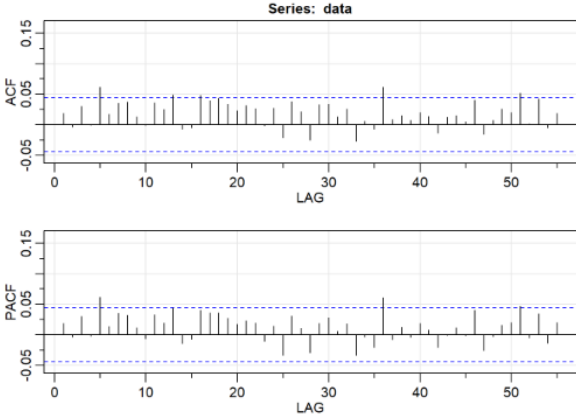
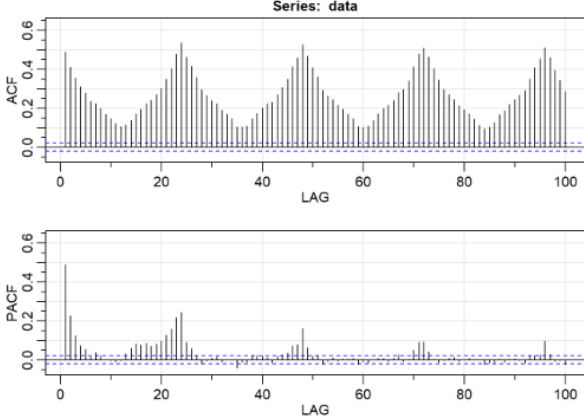
과제명 (Title) : 모델링-6

일자	연구내용	지도 대학원생
12/5	<p>전체 소스코드 : <a href="https://github.com/100yy/lstm_com/blob/master/pred_people_simple_ver.ipynb">https://github.com/100yy/lstm_com/blob/master/pred_people_simple_ver.ipynb</a></p> <p><b>ploting 통한 예측력 확인</b> trainingset에 대한 적합, prediction의 정확도 확인</p> <pre> In [22]: ##예측 def plot_comparison(start_idx, length, train):     if train:         x = x_train         y_true = y_train     else:         x = x_test.reshape(x_test.shape[0], 24*2+14, 1)         y_true = y_test     end_idx = start_idx + length     x = x[start_idx:end_idx]     y_true = y_true[start_idx:end_idx]     y_pred = model.predict(x)     y_pred=scaler.inverse_transform(y_pred)     y_true=scaler.inverse_transform(y_true)      for signal in range(14):         signal_pred = y_pred[:, signal]         signal_true = y_true[:, signal]         plt.figure(figsize=(15,5))         plt.plot(signal_true, label='true')         plt.plot(signal_pred, label='pred')          plt.ylabel("population")         plt.legend()         plt.show()  ##training set에 대해 plot_comparison(0, length=24*28, train=True) ##predict plot_comparison(0, length=24*7, train=False) </pre>  <p><b>모델저장</b></p> <pre> In [24]: from tensorflow.python.keras.models import load_model model.save("LSTM_COM_MODEL_부교1종.h5") </pre>	
기록자:		(서명)

과제명 (Title) : 시계열 분석

일자	연구내용	지도 대학원생
12/9	<p>전체 소스코드 : <a href="https://github.com/100jy/lstm_com/">https://github.com/100jy/lstm_com/</a> longterm_timedependency 100jy 2019 11 16 ACF그림</p> <p>간term의 time lag를 가진 dependency의 존재확인</p> <pre>library("astsa") setwd("C:/Users/mandu/Desktop") 부전1들_유동_1ast&lt;-read.csv("유동인구_부전1들.csv")  #라부과 데이터의 산점(행렬추출) data=부전1들_유동_1ast\$label[1:24] data=datatnorm(24,0,0.1) acf2(data)</pre>  <pre>##          ACF  PACF ## [1,] -0.04 -0.04 ## [2,]  0.04  0.03 ## [3,] -0.08 -0.07 ## [4,] -0.07 -0.14 ## [5,] -0.05 -0.05 ## [6,] -0.15 -0.14 ## [7,] -0.15 -0.03</pre> <pre>#라부과 데이터의 산점(행렬, 유동인구, 산점인출, 것으로, lag0, 부전...) data=부전1들_유동_1ast\$label[1:(24*28)] data=datatnorm(24*28,0,0.1) acf2(data)</pre>  <pre>##          ACF  PACF ## [1,]  0.02  0.02 ## [2,] -0.01 -0.01 ## [3,] -0.03 -0.03 ## [4,] -0.03 -0.03 ## [5,]  0.02  0.02 ## [6,]  0.00 -0.01 ## [7,] -0.02 -0.02 ## [8,]  0.03  0.04 ## [9,]  0.05  0.02 ## [10,] -0.02 -0.02 ## [11,]  0.03  0.04 ## [12,]  0.00  0.01 ## [13,]  0.02  0.02 ## [14,] -0.03 -0.03 ## [15,] -0.04 -0.03 ## [16,]  0.03  0.03</pre>	
기록자:		(서명)

과제명 (Title) : 시계열 분석-2

일자	연구내용	지도 대학원생
12/10	<p>전체 소스코드 : <a href="https://github.com/100yy/lstm_com/">https://github.com/100yy/lstm_com/</a></p>  <pre>##      ACF  PACF ## [1.] 0.02 0.02 ## [2.] 0.00 0.00 ## [3.] 0.03 0.03 ## [4.] 0.00 0.00 ## [5.] 0.06 0.06 ## [6.] 0.02 0.01</pre> <pre>#월별과 데이터의 상관(인term를 가지고 뚜렷한 상관을 보임) data=부진1등_유동_last\$label[1:(24*28*12)] data=data+rnorm(24*12*28,0,0.1) acf2(data)</pre>  <pre>##      ACF  PACF ## [1.] 0.49 0.49 ## [2.] 0.41 0.29 ## [3.] 0.36 0.19 ## [4.] 0.31 0.07 ## [5.] 0.26 0.05 ## [6.] 0.23 0.01 ## [7.] 0.22 0.04 ## [8.] 0.20 0.02</pre>	
	기록자:	(서명)